

**IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD****Publication number:** JP2006518058 (T)**Publication date:** 2006-06-03**Inventor(s):****Applicant(s):****Classification:****- International:** G06F15/80; G06F9/30; G06F9/38; G06F9/445; G06F9/46; G06F15/78; G06F15/76; G06F9/30; G06F9/38; G06F9/445; G06F9/46**- European:** G06F9/38S4**Application number:** JP20050502225T 20031031**Priority number(s):** US20020422503P 20021031; US20030683929 20031009; US20030683932 20031009; US20030684053 20031009; US20030684057 20031009; US20030684102 20031009; WO2003US34558 20031031**Also published as:** WO2004042660 (A2) WO2004042560 (A3) WO2004042574 (A2) WO2004042574 (A3) WO2004042569 (A2)[more >>](#)

Abstract not available for JP 2006518058 (T)

Abstract of corresponding document: WO 2004042560 (A2)

A peer-vector machine includes a host processor and a hardwired pipeline accelerator. The host processor executes a program, and, in response to the program, generates host data, and the pipeline accelerator generates pipeline data from the host data. Alternatively, the pipeline accelerator generates the pipeline data, and the host processor generates the host data from the pipeline data. Because the peer-vector machine includes both a processor and a pipeline accelerator, it can often process data more efficiently than a machine that includes only processors or only accelerators. For example, one can design the peer-vector machine so that the host processor performs decision-making and non-mathematically intensive operations and the accelerator performs non-decision-making and mathematically intensive operations. By shifting the mathematically intensive operations to the accelerator, the peer-vector machine often can, for a given clock frequency, process data at a speed that surpasses the speed at which a processor-only machine can process the data.

---

Data supplied from the [esp@cenet](mailto:esp@cenet) database --- Worldwide

(19) 日本国特許庁 (JP)

## (12) 公 表 特 許 公 報 (A)

(11) 特許出願公表番号

特表2006-518058

(P2006-518058A)

(43) 公表日 平成18年8月3日 (2006.8.3)

(61) Int.Cl.

G06F 15/80 (2006.01)

F1

G06F 15/80

テーマコード (参考)

審査請求 未請求 予備審査請求 未請求 (全 44 頁)

(21) 出願番号 特願2005-502225 (P2005-502225)	(71) 出願人 504242618 ロッキード マーティン コーポレーション アメリカ合衆国 メリーランド州 208 17-1803 ベセスダ ロックランジ ドライブ 6801
(66) (22) 出願日 平成15年10月31日 (2003.10.31)	(74) 代理人 100083932 弁理士 廣江 武典
(65) 翻訳文提出日 平成17年6月16日 (2005.6.16)	(74) 代理人 100129698 弁理士 武川 隆宣
(66) 國際出願番号 PCT/US2003/034558	(74) 代理人 100129676 弁理士 ▲高▼荒 新一
(67) 國際公開番号 WO2004/042562	(74) 代理人 100130074 弁理士 中村 駿元
(61) 優先権主張番号 60/422,503	
(32) 優先日 平成14年10月31日 (2002.10.31)	
(33) 優先権主張国 米国 (US)	
(31) 優先権主張番号 10/683,929	
(32) 優先日 平成15年10月9日 (2003.10.9)	
(33) 優先権主張国 米国 (US)	
(31) 優先権主張番号 10/683,832	
(32) 優先日 平成15年10月9日 (2003.10.9)	
(33) 優先権主張国 米国 (US)	

最終頁に続く

(54) 【発明の名称】 改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法

## (57) 【要約】

パイプライン加速器はメモリと該メモリと結合されたハードウェアに組み込まれたパイプライン回路とを含む。このハードウェアに組み込まれたパイプライン回路は、データを受信し、前記データを前記メモリにロードし、前記メモリから前記データを検索し、前記検索されたデータを処理し、そして前記処理されたデータを外部ソースに提供するように動作できる。加えて或は代替例で、ハードウェアに組み込まれたパイプライン回路は、データを受信し、前記受信されたデータを処理し、前記処理されたデータを前記メモリにロードし、前記メモリから前記処理されたデータを検索し、そして前記検索された処理データを外部ソースに提供するように動作できる。パイプライン加速器がピア-ペクトル・マシンの一部としてのプロセッサと結合されている場合、前記メモリは前記ハードウェアに組み込まれたパイプライン回路と前記プロセッサが実行するアプリケーションとの間ににおけるデータの転送・単向性であろうが二方向性であろうが一を補助する。

## 【特許請求の範囲】

## 【請求項 1】

バイブライン加速器であって、  
メモリと、  
前記メモリと結合されたハードウェアに組み込まれたバイブライン回路と、を含み、  
前記バイブライン回路が、  
データを受信し、  
前記データを前記メモリにロードし、  
前記メモリから前記データを検索し、  
前記検索されたデータを処理し、  
前記処理されたデータを外部ソースに提供するように動作できる、バイブライン加速器。

10

## 【請求項 2】

前記メモリが第1集積回路上に配置され、  
前記バイブライン回路が第2集積回路上に配置されている、請求項1に記載のバイブライン加速器。

## 【請求項 3】

前記バイブライン回路がフィールド・プログラマブル・ゲート・アレイ上に配置されている、請求項1に記載のバイブライン加速器。

20

## 【請求項 4】

前記バイブライン回路が、  
前記処理されたデータを前記メモリにロードし、  
前記メモリから前記処理されたデータを検索し、  
前記検索された処理データを前記外部ソースに提供することによって、前記処理されたデータを前記外部ソースに提供するように動作できる、請求項1に記載のバイブライン加速器。

## 【請求項 5】

前記外部ソースがプロセッサを含み、  
前記バイブライン回路が前記プロセッサから前記データを受信するように動作できる、請求項1に記載のバイブライン加速器。

30

## 【請求項 6】

計算マシンであって、  
プロセッサと、  
前記プロセッサと結合されたバイブライン加速器と、を含み、  
前記バイブライン加速器が、  
メモリと、  
前記メモリと結合されたハードウェアに組み込まれたバイブライン回路と、を含み、  
前記バイブライン回路が、  
前記プロセッサからデータを受信し、  
前記データを前記メモリにロードし、  
前記メモリから前記データを検索し、  
前記検索されたデータを処理し、  
前記処理されたデータを前記プロセッサに提供するように動作できる、計算マシン。

40

## 【請求項 7】

バイブライン加速器であって、  
メモリと、  
前記メモリと結合されたハードウェアに組み込まれたバイブライン回路と、を含み、  
前記バイブライン回路が、  
データを受信し、

50

前記受信されたデータを処理し、  
前記処理されたデータを前記メモリにロードし、  
前記メモリから前記処理されたデータを検索し、  
前記検索された処理データを外部ソースに提供するように動作できる、パイプライン  
加速器。

## 【請求項 8】

計算マシンであって、  
プロセッサと、  
前記プロセッサと結合されたパイプライン加速器と、を含み、  
前記パイプライン加速器が、  
メモリと、  
ハードウェアに組み込まれたパイプライン回路と、を含み、  
前記ハードウェアに組み込まれたパイプライン回路が、  
前記プロセッサからデータを受信し、  
前記受信されたデータを処理し、  
前記処理されたデータを前記メモリにロードし、  
前記メモリから前記処理されたデータを検索し、  
前記検索された処理データを前記プロセッサに提供するように動作できる、計  
算マシン。

## 【請求項 9】

パイプライン加速器であって、  
第1及び第2のメモリと、  
前記第1及び第2のメモリと結合されたハードウェアに組み込まれたパイプライン回路  
と、を含み、  
前記パイプライン回路が、  
外部ソースから生データを受信してその生データを前記第1メモリにロードするよう  
に動作できる入力データ・ハンドラーと、  
前記生データを処理するように動作できるハードウェアに組み込まれたパイプライン  
と、  
前記第1メモリから前記生データを検索し、その検索された生データを前記ハードウ  
エアに組み込まれたパイプラインに提供してから、前記ハードウェアに組み込まれたパイ  
プラインから前記第2メモリに処理されたデータをロードするように動作できるパイプラ  
イン・インターフェースと、  
前記第2メモリから前記処理されたデータを検索してその処理されたデータを前記外  
部ソースに提供するように動作できる出力データ・ハンドラーと、を含む、パイプライン  
加速器。

## 【請求項 10】

前記第1及び第2のメモリの各々がそれぞれの第1及び第2のポートを含み、  
前記入力データ・ハンドラーが、前記第1メモリの前記第1ポートを介して、前記生デ  
ータをロードするように動作でき、  
前記パイプライン・インターフェースが、前記第1メモリの前記第2ポートを介して、  
前記生データを検索して前記処理されたデータを前記第2メモリの前記第1ポートを介し  
てロードするように動作でき、  
前記出力データ・ハンドラーが、前記第2メモリの前記第2ポートを介して、前記処理  
されたデータを検索するように動作できる、請求項9に記載のパイプライン加速器。

## 【請求項 11】

前記ハードウェアに組み込まれたパイプライン回路と結合された第3メモリを更に含み  
&  
前記ハードウェアに組み込まれたパイプラインが、前記生データを処理しながら中間デ  
ータを生成するように動作でき、

10

20

30

40

50

前記バイブルайн・インターフェースが、前記中間データを前記第3メモリにロードして前記第3メモリから前記中間データを検索するように動作できる、請求項9に記載のバイブルайн加速器。

## 【請求項1-2】

前記第1及び第2のメモリが第1及び第2の集積回路上にそれぞれ配置され、

前記バイブルайн回路がフィールド・プログラマブル・ゲート・アレイ上に配置されている、請求項9に記載のバイブルайн加速器。

## 【請求項1-3】

前記入力データ・ハンドラー及び前記バイブルайн・インターフェースと結合されている入力データ・キューを更に含み、

前記入力データ・ハンドラーが、前記第1メモリ内の前記生データの箇所を指すポインタを前記入力データ・キューにロードし、

前記バイブルайн・インターフェースが前記箇所から前記ポインタを用いて前記生データを検索するように動作できる、請求項9に記載のバイブルайн加速器。

## 【請求項1-4】

前記出力データ・ハンドラー及び前記バイブルайн・インターフェースと結合された出力データ・キューを更に含み、

前記バイブルайн・インターフェースが、前記第2メモリ内の前記処理データの箇所を指すポインタを前記出力データ・キューにロードし、

前記出力データ・ハンドラーが、前記箇所から前記ポインタを用いて前記処理データを検索するように動作できる、請求項9に記載のバイブルайн加速器。

## 【請求項1-5】

前記入力データ・ハンドラー、ハードウェアに組み込まれたバイブルайн、バイブルайн・インターフェース、並びに、出力データ・ハンドラーが各動作コンフィギュレーションを有し、

前記入力データ・ハンドラー、ハードウェアに組み込まれたバイブルайн、バイブルайн・インターフェース、並びに、出力データ・ハンドラーと結合されると共に、それらの前記動作コンフィギュレーションを設定するように動作できるコンフィギュレーション・マネージャを更に含む、請求項9に記載のバイブルайн加速器。

## 【請求項1-6】

前記入力データ・ハンドラー、ハードウェアに組み込まれたバイブルайн、バイブルайн・インターフェース、並びに、出力データ・ハンドラーが各動作状況を有し、

前記入力データ・ハンドラー、ハードウェアに組み込まれたバイブルайн、バイブルайн・インターフェース、並びに、出力データ・ハンドラーと結合されると共に、動作状況に応じて、それらにおける例外を識別するように動作できる例外マネージャを更に含む、請求項9に記載のバイブルайн加速器。

## 【請求項1-7】

バイブルайн加速器であって、

データを処理するように動作できるハードウェアに組み込まれたバイブルайнと、

前記ハードウェアに組み込まれたバイブルайнと結合された入力データ・ハンドラーと、を含み、

前記入力データ・ハンドラーが、

前記データを受信し、

前記データが前記ハードウェアに組み込まれたバイブルайнに向けられているかを決定し、

前記データが前記ハードウェアに組み込まれたバイブルайнに向けられていれば、そのデータをそのハードウェアに組み込まれたバイブルайнに提供するように動作できる、バイブルайн加速器。

## 【請求項1-8】

前記入力データ・ハンドラーが、

10

20

30

40

50

ヘッダー及び前記データを含むメッセージを受信して、前記メッセージから前記データを抽出することによって前記データを受信し、

前記ヘッダーを分析することによって前記データが前記ハードウェアに組み込まれたバイブラインに向けられているかを決定するように更に動作できる、請求項17に記載のバイブライン加速器。

【請求項19】

前記ハードウェアに組み込まれたバイブライン及び前記入力データ・ハンドラーが單一のフィールド・プログラマブル・ゲート・アレイ上に配置されている、請求項17に記載のバイブライン加速器。

【請求項20】

前記ハードウェアに組み込まれたバイブライン及び前記入力データ・ハンドラーが各フィールド・プログラマブル・ゲート・アレイ上にそれぞれ配置されている、請求項17に記載のバイブライン加速器。

【請求項21】

計算マシンであって、

プロセッサと、

前記プロセッサと結合されたバイブライン加速器と、を含み、

前記バイブライン加速器が、

データを処理するように動作できるハードウェアに組み込まれたバイブラインと、

前記ハードウェアに組み込まれたバイブラインと結合されると共に、前記プロセッサから前記データを受信し、前記データが前記ハードウェアに組み込まれたバイブラインに向けられているかを決定し、もし前記データが前記ハードウェアに組み込まれたバイブラインに向けられていれば、前記データを前記ハードウェアに組み込まれたバイブラインに提供するように動作できる入力データ・ハンドラーと、含む、計算マシン。

【請求項22】

バイブライン加速器であって、

データを生成するように動作できるハードウェアに組み込まれたバイブラインと、

前記ハードウェアに組み込まれたバイブラインと結合された出力データ・ハンドラーと、を含み、

前記出力データ・ハンドラーが、

前記データを受信し、

前記データの仕向先を決定し、

前記データを前記仕向先に提供するように動作できる、バイブライン加速器。

【請求項23】

前記出力データ・ハンドラーが、

前記データのタイプを識別して、前記データの前記タイプに基づき前記仕向先を決定することによって前記データの前記仕向先を決定するように動作できると共に、

前記仕向先を識別し且つ前記データを含むメッセージを生成して、前記メッセージを前記仕向先に提供するように動作できる、請求項22に記載のバイブライン加速器。

【請求項24】

計算マシンであって、

アプリケーションのスレッドを実行するように動作できるプロセッサと、

前記プロセッサと結合されたバイブライン加速器と、を含み、

前記バイブライン加速器が、

データを生成するように動作できるハードウェアに組み込まれたバイブラインと、

前記ハードウェアに組み込まれたバイブラインと結合された出力データ・ハンドラーと、を含み、

前記出力データ・ハンドラーが、

前記データを受信し、

前記データに加入する前記アプリケーションのスレッドを識別し、

10

20

30

40

50

前記データを前記加入スレッドに提供するように動作できる、計算マシン。

【請求項 2-5】

パイプライン加速器であって、

データ値を処理するように動作できるハードウェアに組み込まれたパイプラインと、

前記ハードウェアに組み込まれたパイプラインと結合されると共に、該ハードウェアに組み込まれたパイプラインの動作を制御するように動作できるシーケンス・マネージャと、を含むパイプライン加速器。

【請求項 2-6】

前記シーケンス・マネージャが前記ハードウェアに組み込まれたパイプラインが前記データ値を受信する順番を翻覆するように動作できる、請求項 2-5 に記載のパイプライン加速器。

10

【請求項 2-7】

前記シーケンス・マネージャが、

事象を受信し、

前記事象に応じて前記ハードウェアに組み込まれたパイプラインを制御するように更に動作できる、請求項 2-6 に記載のパイプライン加速器。

【請求項 2-8】

前記シーケンス・マネージャが、

同期信号を受信し、

前記同期信号に応じて前記ハードウェアに組み込まれたパイプラインの動作を制御する 20 ように更に動作できる、請求項 2-5 に記載のパイプライン加速器。

【請求項 2-9】

前記シーケンス・マネージャが、

前記ハードウェアに組み込まれたパイプラインに対する実現値を検出し、

前記実現値に応じて事象を生成するように更に動作できる、請求項 2-5 に記載のパイプライン加速器。

【請求項 3-0】

計算マシンであって、

データ及び事象を生成するように動作できるプロセッサと、

前記プロセッサと結合されたパイプライン加速器と、を含み、

前記パイプライン加速器が、

前記プロセッサから前記データを受信してその受信されたデータを処理するように動作できるハードウェアに組み込まれたパイプラインと、

前記ハードウェアに組み込まれたパイプラインと結合されると共に、前記プロセッサから前記事象を受信して前記事象に応じて前記ハードウェアに組み込まれたパイプラインの動作を制御するように動作できるシーケンス・マネージャと、を含む、計算マシン。

30

【請求項 3-1】

パイプライン加速器であって、

動作コンフィギュレーションを有すると共にデータを処理するように動作できるハードウェアに組み込まれたパイプライン回路と、

40

前記ハードウェアに組み込まれたパイプライン回路と結合されると共に、前記動作コンフィギュレーションを設定するように動作できるコンフィギュレーション・マネージャと、を含むパイプライン加速器。

【請求項 3-2】

前記ハードウェアに組み込まれたパイプライン回路がコンフィギュレーション・レジスタを含み、

前記コンフィギュレーション・マネージャが前記コンフィギュレーション・レジスタにコンフィギュレーション値をロードすることによって前記動作コンフィギュレーションを設定するように動作できる、請求項 3-1 に記載のパイプライン加速器。

40

【請求項 3-3】

50

前記コンフィギュレーション・マネージャが外部ソースから前記コンフィギュレーション値を受信するように動作できる、請求項3-2に記載のパイプライン加速器。

## 【請求項3-4】

計算マシンであって、  
データ及びコンフィギュレーション値を生成するように動作できるプロセッサと、  
前記プロセッサと結合されたパイプライン加速器と、を含み、  
前記パイプライン加速器が、

動作コンフィギュレーションを有すると共に前記データを処理するように動作できる  
ハードウェアに組み込まれたパイプライン回路と、

前記ハードウェアに組み込まれたパイプライン回路と結合されると共に、前記コンフ  
ィギュレーション値に応じて前記動作コンフィギュレーションを設定するように動作でき  
るコンフィギュレーション・マネージャと、を含む、計算マシン。

10

## 【請求項3-5】

パイプライン加速器であって、  
動作状況を有すると共にデータを処理するように動作できるハードウェアに組み込まれ  
たパイプライン回路と、  
前記ハードウェアに組み込まれた回路と結合されると共に、前記動作状況に応じて前記  
ハードウェアに組み込まれた回路の前記動作状況における例外を識別するように動作でき  
る例外マネージャと、  
を含むパイプライン加速器。

20

## 【請求項3-6】

前記ハードウェアに組み込まれたパイプライン回路が前記動作状況を表す状況値を生成  
するように動作でき、  
前記例外マネージャが前記状況値に応じて前記例外を識別するように動作できる、請求  
項3-5に記載のパイプライン加速器。

## 【請求項3-7】

前記ハードウェアに組み込まれたパイプライン回路が前記状況値を記憶するように動作  
できる状況レジスタを含み、  
前記例外マネージャが前記状況レジスタから前記状況値を受信する、請求項3-6に記載  
のパイプライン加速器。

30

## 【請求項3-8】

前記例外マネージャが、外部ソースへの前記ハードウェアに組み込まれたパイプライン  
回路の前記動作状況における例外を識別するように動作できる、請求項3-5に記載のパイ  
プライン加速器。

## 【請求項3-9】

計算マシンであって、  
データを生成するように動作できるプロセッサと、  
パイプライン加速器と、を含み、  
前記パイプライン加速器が、

動作状況を有すると共にデータを処理して前記動作状況を表す状況値を生成するよう  
に動作できるハードウェアに組み込まれたパイプライン回路と、

40

前記ハードウェアに組み込まれたパイプライン回路と結合されると共に、前記状況値  
に応じて前記ハードウェアに組み込まれたパイプライン回路の前記動作状況における例外  
を識別して前記プロセッサに前記例外を通知するように動作できる例外マネージャと、を  
含む、計算マシン。

## 【請求項4-0】

計算マシンであって、  
動作状況を有してデータを処理するように動作できるハードウェアに組み込まれたパイ  
プライン回路と、前記ハードウェアに組み込まれたパイプライン回路と結合されると共に、  
前記動作状況を表す状況値を生成するように動作できる例外マネージャと、を含むパイ

50

ライイン加速器と、

前記バイライイン加速器と結合されると共に、前記データを生成し、前記状況値を受信してから、前記状況値を分析することによって前記ハードウェアに組み込まれたバイライイン回路が動作しているかを決定するように動作できるプロセッサと、を含む、計算マシン。

【請求項 4 1】

方法であつて、  
データをメモリにロードし、  
前記メモリから前記データを検索し、  
ハードウェアに組み込まれたバイライイン回路によって前記検索されたデータを処理し 10  
'、  
前記処理されたデータを外部ソースに提供することを含む方法。

【請求項 4 2】

前記処理されたデータを提供することが、  
前記処理されたデータを前記メモリにロードし、  
前記メモリから前記処理されたデータを検索し、  
前記検索された処理データを前記外部ソースに提供することを含む、請求項 4 1 に記載の方法。

【請求項 4 3】

方法であつて、  
ハードウェアに組み込まれたバイライイン回路によってデータを処理し、  
前記処理されたデータをメモリにロードし、  
前記メモリから前記処理されたデータを検索し、  
前記検索された処理データを外部ソースに提供することを含む、方法。 20

【請求項 4 4】

方法であつて、  
外部ソースから第 1 メモリに生データをロードし、  
前記第 1 メモリから前記生データを検索し、  
ハードウェアに組み込まれたバイライインによって前記検索されたデータを処理し、  
前記ハードウェアに組み込まれたバイライインから第 2 メモリに前記処理されたデータ 30  
をロードし、  
前記第 2 メモリから前記外部ソースに前記処理されたデータを提供することを含む、方法。

【請求項 4 5】

前記生データをロードすることが、前記第 1 メモリの第 1 ポートを介して前記生データをロードすることを含み、  
前記生データを検索することが、前記第 1 メモリの第 2 ポートを介して前記生データを検索することを含み、  
前記処理されたデータをロードすることが、前記第 2 メモリの第 1 ポートを介して前記処理されたデータをロードすることを含み、  
前記処理されたデータを提供することが、前記第 2 メモリの第 2 ポートを介して前記処理されたデータを検索することを含む、請求項 4 4 に記載の方法。 40

【請求項 4 6】

前記生データの処理に応じて前記ハードウェアに組み込まれたバイライインによって中間データを生成し、  
前記中間データを第 3 メモリにロードし、  
前記第 3 メモリから前記ハードウェアに組み込まれたバイライインに依るよう前記中間データを提供することを含む、請求項 4 4 に記載の方法。

【請求項 4 7】

前記第 1 メモリ内の前記生データの箇所を指すポインタを人力メッセージ・キューにロ 50

ードすることを更に含み、

前記生データを検索することが、前記箇所から前記ポインタを用いて前記生データを検索することを含む、請求項4-4に記載の方法。

【請求項4-8】

前記第2メモリ内の前記処理されたデータの箇所を指すポインタを出力メッセージ・キーにコードすることを更に含み、

前記処理されたデータを検索することが、前記箇所から前記ポインタを用いて前記処理されたデータを検索することを含む、請求項4-4に記載の方法。

【請求項4-9】

前記生データをロードして検索し、その検索されたデータを処理して、その処理されたデータをロードして提供するためのパラメータを設定することを更に含む、請求項4-4に記載の方法。

10

【請求項5-0】

前記生データの前記ロード及び前記検索、前記検索されたデータの前記処理、並びに、前記処理されたデータの前記ロード及び前記提供の間にエラーが生ずるかを決定することを更に含む、請求項4-4に記載の方法。

【請求項5-1】

方法であって、  
データを受信し、  
前記データがハードウェアに組み込まれたパイプラインに向けられているかを決定し、  
もし前記データが前記ハードウェアに組み込まれたパイプラインに向けられていれば前記ハードウェアに組み込まれたパイプラインに前記データを提供することを含む方法。

20

【請求項5-2】

前記データを受信することが、  
ヘッダー及び前記データを含むメッセージを受信し、  
前記メッセージから前記データを抽出することを含み、  
前記データが前記ハードウェアに組み込まれたパイプラインに向けられているかを決定することが、前記ヘッダーを分析することを含む、請求項5-1に記載の方法。

30

【請求項5-3】

方法であって、  
ハードウェアに組み込まれたパイプラインによってデータを生成し、  
前記データの仕向先を決定し、  
前記データを前記仕向先に提供することを含む方法。

【請求項5-4】

前記データの前記仕向先を決定することが、  
前記データのタイプを識別し、  
前記データの前記タイプに基づき前記仕向先を決定することを含み、  
前記データを前記仕向先に提供することが、  
前記仕向先を識別すると共に前記データを含むメッセージを生成し、  
前記メッセージを前記仕向先に提供することを含む、請求項5-3に記載の方法。

40

【請求項5-5】

方法であって、  
ハードウェアに組み込まれたパイプラインによってデータ値を処理し、  
前記ハードウェアに組み込まれたパイプラインの動作を順序付けすることを含む方法。

【請求項5-6】

前記動作を順序付けることが、  
前記ハードウェアに組み込まれたパイプラインが前記データ値を処理する順序を順序付けることを含む、請求項5-5に記載の方法。

【請求項5-7】

前記動作を順序付けることが、

50

前記ハードウェアに組み込まれたパイプラインを同期信号と同期させることを含む、請求項 5-5 に記載の方法。

【請求項 5-8】

前記ハードウェアに組み込まれたパイプラインの動作中に予め規定された実現値を検知し、

前記実現値に応じて事象を生成することを更に含む、請求項 5-5 に記載の方法。

【請求項 5-9】

方法であって、

コンフィギュレーション値をレジスタにロードし、

前記コンフィギュレーション値によってハードウェアに組み込まれたパイプラインの動作コンフィギュレーションを設定することを含む方法。 10

【請求項 6-0】

方法であって、

ハードウェアに組み込まれたパイプラインによってデータを処理し、

前記ハードウェアに組み込まれたパイプラインの動作状況を分析することによって前記処理されたデータにおけるエラーを識別することを含む方法。

【請求項 6-1】

ハードウェアに組み込まれたパイプライン回路を設計する方法であって、

ライブラリから通信インターフェースの第 1 データ表現を検索し、

前記通信インターフェースと結合されることになるハードウェアに組み込まれたパイプラインの第 2 データ表現を生成し、 20

前記第 1 及び第 2 のデータ表現を組み合わせて、前記ハードウェアに組み込まれたパイプライン回路に対するハード・コンフィギュレーション・データを生成することを含む方法。

【請求項 6-2】

前記第 1 及び第 2 のデータ表現を組み合わせる前に、前記サービス層の所定パラメータに対する値を選択することによって前記第 1 データ表現を変更することを更に含む、請求項 6-1 に記載の方法。

【請求項 6-3】

前記通信インターフェースが、前記ハードウェアに組み込まれたパイプライン回路に別の回路と通信させるように動作できる、請求項 6-1 に記載の方法。 30

【請求項 6-4】

前記第 1 及び第 2 のデータ表現を組み合わせることが、前記第 1 及び第 2 のデータ表現を前記ハード・コンフィギュレーション・データに組み合わせるを含む、請求項 6-1 に記載の方法。

【請求項 6-5】

前記ハード・コンフィギュレーション・データがファームウェアを含む、請求項 6-1 に記載の方法。

【発明の詳細な説明】

【技術分野】

【0001】

<優先権の請求>

この出願は、下記の特許文献 1 に対する優先権を請求するものであり、引用することここで合体させる。

【特許文献 1】米国仮出願第 60/422,503 号 (2002 年 10 月 31 日出願)

【0002】

<関連出願の相互参照>

この出願は、「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された下記の特許文献 2、「改善された計算アーキテクチャを有する計算マシン、関連システム、並びに、方法」と題された下記の特許文献 3、「プログラマブル回路、関連計算マ 50

シン、並びに、方法」と題された下記の特許文献4、「多数バイブルайн・ユニットを有するバイブルайн加速器、関連計算マシン、並びに、方法」と題された下記の特許文献5と関連し、これら特許文献は全て2003年10月9日に出願され、其の所有者を有し、引用することとここに合体させる。

【特許文献2】米国出願第10/684,102号

【特許文献3】米国出願第10/684,053号

【特許文献4】米国出願第10/684,057号

【特許文献5】米国出願第10/683,932号

【背景技術】

【0003】

比較的大量のデータを比較的短い期間で処理する通常の計算アーキテクチャは、処理負担を分担する多数の相互接続プロセッサを含む。処理負担を分担することによって、これら多数のプロセッサは、しばしば、所与のクロック周波数で单一プロセッサができるものよりも迅速にデータを処理できる。例えば、これらプロセッサの各々はデータの各部分を処理できるか、或は、処理アルゴリズムの各部分を実行できる。

【0004】

図1は、多数プロセッサ・アーキテクチャを有する従来の計算マシン10の概略ブロック図である。この計算マシン10は、マスター・プロセッサ12と、相互に通信すると共に該マスター・プロセッサとバス16を介して通信する共同プロセッサ14,14と、遠隔装置(図1では不図示)から生データを受け取る入力ポート18と、遠隔装置に処理データを提供する出力ポート20とを含む。また、計算マシン10はマスター・プロセッサ12に対するメモリ22と、共同プロセッサ14,14に対する各メモリ24,24と、マスター・プロセッサ及び共同プロセッサがバス16を介して共有するメモリ26とを含む。メモリ22はマスター・プロセッサ12に対するプログラム及び作業メモリの双方の役割を果たし、各メモリ24,24は各共同メモリ14,14に対するプログラム及び作業メモリの双方の役割を果たす。共有されたメモリ26は、マスター・プロセッサ12及び共同プロセッサ14がそれらの間でデータを転送すること、ポート18を介して遠隔装置からデータを転送すること、ポート20を介して遠隔装置にデータを転送することを可能としている。またマスター・プロセッサ12及び共同プロセッサ14は、マシン10が生データを処理する速度を制御する共通クロック信号を受け取る。

【0005】

一般に、計算マシン10は、マスター・プロセッサ12及び共同プロセッサ14の間で生データの処理を効果的に分割する。ソナー、アレイ等の遠隔ソース(図1では不図示)は、ポート18を介して、生データに対する先入れ先出し(FIFO)バッファ(不図示)として作用する共有メモリ26の1つの区分に生データをロードする。マスター・プロセッサ12はバス16を介してメモリ26から生データを検索して、マスター・プロセッサ及び共同プロセッサ14はその生データを処理して、バス16を介して必要に応じてデータをそれらの間に転送する。マスター・プロセッサ12はその処理データを共有メモリ26内に規定された別のFIFOバッファ(不図示)にロードし、遠隔ソースがポート20を介してこのFIFOからその処理データを検索する。

【0006】

演算例において、計算マシン10は生データに対するn+1個の各演算を順次実行することによって該生データを処理し、これら演算は一體的に高速フーリエ変換(FFT)等の処理アルゴリズムを構成する。より詳細には、マシン10はマスター・プロセッサ12及び共同プロセッサ14からのデータ・処理バイブルайнを形成する。クロック信号の所与の周波数で、そうしたバイブルайнはしばしばマシン10が单一プロセッサのみを有するマシンよりも高速に生データを処理することを可能としている。

【0007】

メモリ26内における生データFIFO(不図示)からの生データ検索後、マスター・プロセッサ12はその生データに対して三角関数等の第1幕演算を実行する。この演算は

10

20

30

40

第1番結果を生み出し、それをプロセッサ12がメモリ26内に規定された第1番結果F1FO(不図示)に記憶する。典型的には、プロセッサ12はメモリ22内に記憶されたプログラムを実行し、そのプログラムの制御の下で上述した動作を実行する。プロセッサ12はメモリ22を作業メモリとしても使用し得て、当該プロセッサが第1番演算の中間期間に生成するデータを一時的に記憶する。

## 【0008】

次に、メモリ26内における第1番結果F1FO(不図示)からの第1番結果検索後、共同プロセッサ14<sub>1</sub>はその第1番結果に対して対数関数等の第2番演算を実行する。この第2番演算は第2番結果を生み出し、それを共同プロセッサ14<sub>1</sub>がメモリ26内に規定された第2番結果F1FO(不図示)に記憶する。典型的には、共同プロセッサ14<sub>1</sub>はメモリ24<sub>1</sub>内に記憶されたプログラムを実行し、そのプログラムの制御の下で上述した動作を実行する。共同プロセッサ14<sub>1</sub>はメモリ24<sub>1</sub>を作業メモリとしても使用し得て、当該共同プロセッサが第2番演算の中間期間に生成するデータを一時的に記憶する。

10

## 【0009】

次に共同プロセッサ24<sub>2</sub>～24<sub>n</sub>は、共同プロセッサ24<sub>1</sub>に対して先に議論されたものと同様に、(第2番結果～第(n-1)番)結果に対して(第3番演算～第n番)演算を順次実行する。

## 【0010】

共同プロセッサ24<sub>n</sub>によって実行される第n番演算は最終結果、即ち処理データを生み出す。共同プロセッサ24<sub>n</sub>はその処理データをメモリ26内に規定された処理データF1FO(不図示)内にロードし、遠隔装置(図1では不図示)がこのF1FOからその処理データを検索する。

20

## 【0011】

マスター・プロセッサ12及び共同プロセッサ14は処理アルゴリズムの種々の演算を同時に実行するので、計算マシン10は、しばしば、種々の演算を順次実行する单一プロセッサを有する計算マシンよりも生データを高速に処理することができる。詳細には、单一プロセッサは、生データから成る先行集合に対する全(n+1)個の演算を実行するまで、生データから成る新しい集合を検索できない。しかし、以上に議論したバイブライン技術を用いて、マスター・プロセッサ12は第1演算だけを実行後に生データから成る新しい集合を検索できる。結果として、所与のクロック周波数でのバイブライン技術は、单一プロセッサ・マシン(図1では不図示)と比較して約n+1倍だけマシン10が生データを処理する速度を増大することができる。

30

## 【0012】

代替的には、計算マシン10は、生データに対するFFT等の処理アルゴリズムの(n+1)例を同時に実行することによって該生データを並列して処理し得る。即ち、もしもそのアルゴリズムが先行する例において先に記載されたような(n+1)個の順次演算を含めば、マスター・プロセッサ12及び共同プロセッサ14の各々は生データからそれぞれが成る各集合に対して、順次、全(n+1)個の演算を実行する。その結果として、所与のクロック周波数で、先のバイブライン技術と同様のこの並列処理技術は、单一プロセッサ・マシン(図1では不図示)と比較して約n+1倍だけマシン10が生データを処理する速度を増大することができる。

40

## 【0013】

残念ながら、計算マシン10は单一プロセッサ・計算マシン(図1では不図示)と比べてより迅速にデータを処理できるが、マシン10のデータ処理速度はしばしばプロセッサ・クロックの周波数より非常に小さい。詳細には、計算マシン10のデータ処理速度はマスター・プロセッサ12及び共同プロセッサ14がデータ処理するのに必要な時間によって制限される。簡略化のため、この速度制限の例はマスター・プロセッサ12と連携して議論されているが、この議論は共同プロセッサ14にも適用されることを理解して頂きたい。先に議論されたように、マスター・プロセッサ12は所望の方式でデータを操作すべくプロセッサを制御するプログラムを実行する。このプログラムはプロセッサ12が実行

50

する複数の命令から成るシーケンスを含む。残念ながら、プロセッサ12は典型的には單一命令を実行するために多数のクロック・サイクルを必要とし、そしてしばしばデータの單一値を処理すべく多数の命令を実行しなければならない。例えば、プロセッサ12が第1データ値A(不図示)を第2データ値B(不図示)で乗算することを仮定する。第1クロック・サイクル中、プロセッサ12はメモリ22から乗算命令を検索する。第2及び第3クロック・サイクル中、プロセッサ12はメモリ26からA及びBをそれぞれ検索する。第4クロック・サイクル中、プロセッサ12はA及びBを乗算し、そして第5クロック・サイクル中に結果としての積をメモリ22或は26に記憶するか、或は、その結果としての積を遠隔装置(不図示)に提供する。これは最良ケースのシナリオであり、その理由は多くの場合にプロセッサ12はカウンタの初期化及び閉鎖等のオーバーヘッド・タスクに対して付加的なクロック・サイクルを必要とするからである。それ故に、よくてもプロセッサ12はA及びBを処理すべく5クロック・サイクルを必要とするか、或は、1データ値当たり平均2.5クロック・サイクルを必要とする。

## 【0014】

結果として、計算マシン10がデータを処理する速度は、しばしば、マスター・プロセッサ12及び共同プロセッサ14を駆動するクロックの周波数より非常に低い。例えば、もしプロセッサ12は1.0ギガヘルツ(GHz)でクロックされるが、1データ値当たり平均2.5クロック・サイクルを必要とすれば、効果的なデータ処理速度は $(1.0\text{ GHz})/2.5 = 0.4\text{ GHz}$ と同等である。この効果的なデータ処理速度は、しばしば、1秒当たり演算数の単位で特徴付けされる。それ故に、この例において、1.0GHzのクロック速度で、プロセッサ12は0.4ギガ演算数/秒(Gops)で使用限界が定められる。

## 【0015】

図2は、所与クロック周波数で且つしばしば該バイブラインがクロックされる速度と略同一速度で、プロセッサが可能であるよりは高速で典型的にはデータを処理できるハードウェアに組み込まれたデータ・バイブライン30のプロック線図である。バイブライン30は、プログラム命令を実行することなく、各データに対する各演算を各自が実行する演算子回路32<sub>1</sub>～32<sub>n</sub>を含む。即ち、所望の演算は回路32内に「書き込み」が為され、それがプログラム命令の必要性なしに自動的にその演算を実現化するよう為す。プログラム命令の実行と関連されたオーバーヘッドを減することによって、バイブライン30は所与のクロック周波数でプロセッサが可能であるよりは単位秒当たりより多くの演算を典型的には実行する。

## 【0016】

例えば、バイブライン30は所与のクロック周波数でプロセッサが可能であるよりは高速で以下の式 $Y(x_1) = (5x_1 + 3)2^{11}$ をしばしば解くことができる。

$$Y(x_1) = (5x_1 + 3)2^{11}$$

ここで、 $x_1$ は複数の生データ値から成るシーケンスを表す。この例において、演算子回路32<sub>1</sub>は $5x_1$ を計算する乗算器であり、回路32<sub>2</sub>は $5x_1 + 3$ を計算する加算器であり、そして回路32<sub>n</sub>(n=3)は $(5x_1 + 3)2^{11}$ を計算する乗算器である。

## 【0017】

第1クロック・サイクルk=1中、回路32<sub>1</sub>はデータ値 $x_1$ を受け取って、それを5で乗じて、 $5x_1$ を生成する。

## 【0018】

第2クロック・サイクルk=2中、回路32<sub>2</sub>は回路32<sub>1</sub>から $5x_1$ を受け取って、3を加えて、 $5x_1 + 3$ を生成する。またこの第2クロック・サイクル中に回路32<sub>n</sub>は $5x_1$ を生成する。

## 【0019】

第3クロック・サイクルk=3中、回路32<sub>n</sub>は回路32<sub>2</sub>から $5x_1 + 3$ を受け取って、 $2^{11}$ で乗じて(効果的としては、 $x_1$ だけ $5x_1 + 3$ を左シフトする)、第1結果( $5x_1 + 3)2^{11}$ を生成する。またこの第3クロック・サイクル中に回路32<sub>1</sub>は $5x_1$ を生成

10

20

30

40

50

し、回路3-2<sub>z</sub>は $5 \times z + 3$ を生成する。

【0020】

このようにしてバイブライン3-0は、全ての生データ値が処理されるまで、引き続く生データ値 $x_k$ の処理を続行する。

【0021】

結果として、生データ値 $x_1$ の受け取り後の2つのクロック・サイクルの遅延、即ち、この遅延はバイブライン3-0の待ち時間としばしば呼称され、バイブラインは結果 $(5 \times 1 + 3) \cdot 2^{z^1}$ を生成し、その後、1つの結果を生成する、即ち各クロック・サイクル毎に $(5 \times z + 3) \cdot 2^{z^2}, (5 \times z + 3) \cdot 2^{z^3}, \dots, (5 \times z + 3) \cdot 2^{z^k}$ を生成する。

【0022】

待ち時間を無視して、バイブライン3-0はこうしてクロック速度と同等のデータ処理速度を有する。比較して、マスター・プロセッサ1-2及び共同プロセッサ1-4(図1)が先の例におけるようにクロック速度の0.4倍であるデータ処理速度を有すると仮定すれば、バイブライン3-0は、所々のクロック速度で、計算マシン1-0(図1)よりも2.5倍高速でデータを処理できる。

【0023】

更に図2で参照されるように、設計者はフィールド・プログラマブル・ゲート・アレイ(FPGA)等のプログラマブル・ロジックIC(PLIC)にバイブライン3-0を具現化することを選択する可能性があり、その理由はPLICが特殊用途IC(ASIC)が為すよりも多くの設計及び変更の柔軟性を許容するからである。PLIC内にハードウェアに組み込まれた接続を構成するため、設計者はPLIC内に配備された相互接続構成レジスタを單に所定バイナリ状態に設定する。全てのこうしたバイナリ状態の組み合わせはしばしば「ファームウェア」と呼称される。典型的には、設計者はこのファームウェアをPLICと結合された不揮発性メモリ(図2では不図示)内にロードする。PLICを「ターンオン」すると、それはファームウェアをそのメモリから相互接続構成レジスタにダウンロードする。それ故に、PLICの機能を変更すべく、設計者は單にそのファームウェアを変更して、PLICがその変更されたファームウェアを相互接続構成レジスタにダウンロードすることを可能とする。ファームウェアを單に変更することによってPLICを変更する能力は、モード作成段階中や「フィールド内」にバイブライン3-0をアップグレードするために特に有用である。

【0024】

残念ながら、ハードウェアに組み込まれたバイブライン3-0は重要な意思決定、特に入れ子意思決定を引き起こすアルゴリズムを実行すべき最良の選択でない可能性がある。プロセッサは、典型的には、入れ子意思決定命令(例えば、「もしAであれば、Bを為し、またもしCであれば、Dを為し、...、またnを為し等々」のように、入れ子条件命令)を、比較する技術の演算命令(例えば、「A+B」)を実行できる程に高速に実行できる。しかしバイブライン3-0は、比較的単純な決定(例えば、「A>B?」)を効率的に為し得るが、典型的にはプロセッサができる程に効率的に入れ子決定(例えば、「もしAであれば、Bを為し、またもしCであれば、Dを為し、...、またnを為す」)を為すことができない。この非効率性の1つの理由は、バイブライン3-0はほんの僅かなオンボード・メモリしか持たないことがあり、したがって外部作業/プログラム・メモリ(不図示)にアクセスすることを必要とすることがあるからである。そして、こうした入れ子決定を実行すべくバイブライン3-0を設計することができるが、必要とされる回路のサイズ及び複雑性はしばしばそうした設計を非現実的に為し、特にアルゴリズムが多数の種々の入れ子決定を含む場合でそうである。

【0025】

結果として、プロセッサは典型的には重要な意思決定を必要とする用途において使用され、ハードウェアに組み込まれたバイブラインは殆ど意思決定が為されないか或は意思決定されない「ナンバークランチング(数値データ処理)」用途に典型的には限定される。

【0026】

10

20

30

40

50

更には、下記に議論されるように、典型的には、特にパイプライン30が多数のPLCを含む場合、図2のパイプライン等のハードウェアに組み込まれたパイプラインを設計／変更するよりも、図1の計算マシン10等のプロセッサに基づく計算マシンを設計／変更することが非常に易しい。

## 【0027】

プロセッサ及びそれらの周辺機器（例えば、メモリ）等の計算構成要素は、典型的には、プロセッサに基づく計算マシンを形成すべくそれら構成要素の相互接続を補助する工業規格通信インターフェースを含む。

## 【0028】

典型的には、規格通信インターフェースは2つの層、即ち、物理層及びサービス層を含む。

## 【0029】

物理層は、回路とこの回路のインターフェース及び動作パラメータを形成する対応回路相互接続とを含む。例えば、物理層はそれら構成要素を1つのバスに接続するピンと、それらのピンから受け取ったデータをラッチするバッファと、信号をそれらピンに駆動するドライバとを含む。動作パラメータは、ピンが受け取るデータ信号の許容可能電圧範囲と、データの書き込み及び読み取りのための信号タイミングと、動作の支援されたモード（例えば、バーストモード、ページモード）とを含む。従来の物理層はトランジスタ-トランジスタ論理（TTL）及びRAMBUSを含む。

## 【0030】

サービス層は、計算構成要素のデータ転送のためのプロトコルを含む。このプロトコルはデータのフォーマットと、構成要素によるフォーマット済みデータの送受信の方式とを含む。従来の通信プロトコルは、ファイル転送プロトコル（FTP）及び伝送制御プロトコル／インターネット・プロトコル（TCP/IP）を含む。

## 【0031】

結果として、製造業者やその他は工業規格通信インターフェースを有する計算構成要素を典型的には設定するので、そうした構成要素のインターフェースを典型的には設計できて、それを他の計算構成要素と比較的少ない労力で相互接続することができる。これは、計算マシンの他の部分の設計に設計者自信の時間を殆ど費やすことを可能として、各種構成要素を追加或は除去することによってそのマシンを変更することを可能としている。

## 【0032】

工業規格通信インターフェースを支援する計算構成要素を設計することは、設計ライブラリから既存の物理層を用いることによって設計時間を節約することを可能としている。これは、設計者が構成要素を既製の計算構成要素と容易にインターフェースすることを保証するものである。

## 【0033】

そして、共通した工業規格通信インターフェースを支援する計算構成要素を用いる計算マシンを設計することは、設計者がそれら構成要素を少しの時間及び労力で相互接続することを可能としている。それら構成要素は共通インターフェースを支援するので、設計者はそれらをシステム・バスを介して少しの設計労力で相互接続することができる。そして、その支援されたインターフェースは工業規格であるので、マシンを容易に変更することができる。例えば、システム設計が進化するに伴って種々の構成要素及び周辺機器をマシンに追加することができるか、或は、テクノロジーが進化するに伴って次世代の構成要素を追加／設計することができる。更には、構成要素が通常の工業規格サービス層を支援するので、計算マシンのソフトウェアに対応するプロトコルを具現化する既存のソフトウェア・モジュールを組み込むことができる。それ故に、インターフェース設計が本質的には既に整っているので少しの労力で構成要素をインターフェースでき、よって、マシンに所望の機能を実行させるマシンの各種部分（例えばソフトウェア）の設計に集中することができる。

## 【0034】

10

20

30

40

50

しかし残念ながら、図2のバイブライン30等のハードウェアに組み込まれたバイブルインを形成すべく、使用されるPLIC等の各種構成要素に対する既知の工業規格サービス層が全くない。

#### 【0035】

結果として、多数のPLICを有するバイブルインを設計すべく、多大な時間を費やし、「ゼロから」種々のPLICの間の通信インターフェースのサービス層を設計し且つデバッグする多大な労力を行使する。典型的には、そうしたその場限りのサービス層は種々のPLIC間で転送されるデータのパラメータに依存する。同じように、プロセッサとインターフェースするバイブルインを設計すべく、ゼロからのバイブルイン及びプロセッサの間の通信インターフェースのサービス層の設計及びデバッグに関して多大な時間を費やし且つ多大な労力を行使する必要がある。

10

#### 【0036】

同様に、そうしたバイブルインをPLICを該バイブルインに追加することによって変更すべく、典型的には、その追加されたPLICと既存のPLICとの間の通信インターフェースのサービス層の設計及びデバッグに関して多大な時間を費やし且つ多大な労力を行使する。同じように、プロセッサを追加することによってバイブルインを変更すべく、或は、バイブルインを追加することによって計算マシンを変更すべく、バイブルイン及びプロセッサの間の通信インターフェースのサービス層の設計及びデバッグに関して多大な時間を費やし且つ多大な労力を行使しなければならないであろう。

20

#### 【0037】

結果として、図1及び図2で参照されるように、多数のPLICをインターフェースすることとプロセッサをバイブルインにインターフェースすることとの難しさのため、計算マシンを設計する際に多大な妥協を為すことがしばしば強いられる。例えば、プロセッサに基づく計算マシンでは、ナンバークランチング速度を、複雑な意思決定を為す能力に対する設計／変更の柔軟性と交換することを強いられる。逆に、ハードウェアに組み込まれたバイブルインに基づく計算マシンでは、複雑な意思決定を為す能力と設計／変更の柔軟性を、ナンバークランチング速度と交換することを強いられる。更には、多数のPLICをインターフェースすることに掛かる難しさのため、少数のPLICよりも多くのPLICを有するバイブルインに基づくマシンを設計することはしばしば実際的ではない。その結果、実際的なバイブルインに基づくマシンはしばしば制限された機能しか有さない。そして、プロセッサをPLICとインターフェースすることに関する難しさのため、プロセッサを1つのPLICよりも多くのPLICにインターフェースすることは実際的ではない。その結果、プロセッサ及びバイブルインを組み合わせることによって獲得される利益は最少となる。

30

#### 【発明の開示】

#### 【発明が解決しようとする課題】

#### 【0038】

それ故に、プロセッサに基づくマシンの意思決定を為す能力を、ハードウェアに組み込まれたバイブルインに基づくマシンのナンバークランチング速度と組み合わせることを可能とする新しい計算アーキテクチャに対する要望が生じてきている。

40

#### 【課題を解決するための手段】

#### 【0039】

本発明の実施例に従えば、バイブルイン加速器はメモリと該メモリと結合されたハードウェアに組み込まれたバイブルイン回路とを含む。ハードウェアに組み込まれたバイブルイン回路は、データを受信し、該データをメモリにロードし、該メモリからのデータを検索し、その検索データを処理して、その処理されたデータを外部ソースに提供するように動作できる。

#### 【0040】

本発明の別の実施例に従えば、ハードウェアに組み込まれたバイブルイン回路は、データを受信し、その受信されたデータを処理し、その処理されたデータをメモリにロードし

50

、メモリからその処理されたデータを検索して、その検索された処理データを外部ソースに提供するように動作できる。

【0041】

パイプライン加速器がピア-ペクトル・マシンの一部としてのプロセッサと結合されている場合、メモリはハードウェアに組み込まれたパイプライン回路とプロセッサが実行するアプリケーションとの間でのデータの転送（単向性であろうが二方向性であろうが）を補助する。

【発明を実施するための最も良の形態】

【0042】

図3は、本発明の一実施例に従ったピア-ペクトル・アーキテクチャを行うする計算マシン40の概略ブロック線図である。ホストプロセッサ42に加えて、ピア-ペクトル・マシン40はパイプライン加速器44を含み、それがデータ処理の少なくとも一部を実行して、図1の計算マシン10における共同プロセッサ14の列と効率的に置き換わる。それ故に、ホストプロセッサ42及び加速器44（又は以下に議論されるようにそのユニット）はデータ・ペクトルを前後に転送できる「ピア」である。加速器44はプログラム命令を実行しないので、所与のクロック周波数で共同プロセッサの列ができるものよりも著しく高速にデータに対して数学的に集中的な演算を典型的には実行する。結果として、プロセッサ42の意思決定能力と加速器44のナンバーランキング能力とを組み合わせることによって、マシン40はマシン10等の従来の計算マシンと同一の能力を有するが、しばしばそれよりもデータをより高速に処理することができる。更には、以下に議論されるように、加速器44にホストプロセッサ42の通信インターフェースと互換性がある通信インターフェースを設けることが、特にプロセッサの通信インターフェースが工業規格である場合に、マシン40の設計及び変更を補助する。そして、加速器44が多数のパイプライン・ユニット（例えば、PLI Cに基づく回路）を含む場合、それら各ユニットに同一の通信インターフェースを設けることが、特にそれら通信インターフェースが工業規格インターフェースと互換性がある場合に、当該加速器の設計及び変更を補助する。更には、マシン40は以下に議論されると共に先行して引用された特許出願におけるような他の長所等をも提供し得る。

【0043】

更に図3で参照されるように、ホストプロセッサ42及びパイプライン加速器44に加えて、ピア-ペクトル・計算マシン40は、プロセッサ・メモリ46、インターフェース・メモリ48、バス50、ファームウェア・メモリ52、任意選択的な生データ入力ポート54、処理済みデータ出力ポート58、並びに、任意選択的なルータ61を含む。

【0044】

ホストプロセッサ42は処理ユニット62及びメッセージ・ハンドラー64を含み、プロセッサ・メモリ46は処理ユニット・メモリ66及びハンドラー・メモリ68を含み、そのそれぞれがプロセッサ・ユニット及びメッセージ・ハンドラーに対するプログラム及び作業の両メモリとして役立っている。プロセッサ・メモリ46は、加速器コンフィギュレーション・レジストリ70及びメッセージ・コンフィギュレーション・レジストリ72をも含み、それらが、ホストプロセッサ42が加速器44の機能を構成すると共に、該ホストプロセッサ42がメッセージ・ハンドラー64が送信及び受信するメッセージのフォーマットを構成することを可能とするそれぞれのコンフィギュレーション・データを記憶する。

【0045】

パイプライン加速器44は少なくとも1つのPLI C（不圖示）上に配置され、プログラム命令を実行することなしに各データを処理するハードウェアに組み込まれたパイプライン74<sub>1</sub>～74<sub>n</sub>を含む。ファームウェア・メモリ52は加速器44に対するコンフィギュレーション・ファームウェアを記憶する。もしも加速器44が多数のPLI C上に配置されたら、それらPLI C及びそれらの各ファームウェア・メモリは多数パイプライン・ユニット内に配置され得る（図4）。加速器44及びパイプライン・ユニットは、以下に議

論されると共に、先行して引用された「多数バイブライン・ユニットを有するバイブライン加速器、関連計算マシン、並びに、方法」と題された特許文献5に更に議論されている。代替的には、加速器44は少なくとも1つのASIC上に配置され得て、よって構成不可能な内部相互接続を有し得る。この代替例において、マシン40はファームウェア・メモリ52を省略し得る。更には、加速器44が多数バイブライン74を含んで示されているが、ただ1つのバイブラインを含み得る。加えて、図示されていないが、加速器44はディジタル信号プロセッサ(DSP)等の1つ或はそれ以上のプロセッサを含み得る。更には、図示されていないが、加速器44はデータ入力ポート及び/或はデータ出力ポートを含み得る。

## 【0046】

10

ピア-ベクトル・マシン40の一般動作は、先行して引用された「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された特許文献2に議論されており、バイブライン加速器44の構造及び動作は図4乃至図9と連携された以下に議論されている。

## 【0047】

図4は、本発明の一実施例に従った図3のバイブライン加速器44の概略プロック線図である。

## 【0048】

20

加速器44は1つ或はそれ以上のそうしたバイブライン・ユニット78を含み、それらの各々はPLTC或はASIC等のバイブライン回路80を含む。以下で更に議論されると共に先行して引用された「多数バイブライン・ユニットを有するバイブライン加速器、関連計算マシン、並びに、方法」と題された特許文献5にあるように、各バイブライン・ユニット78はホストプロセッサ42の「ピア」であると共に加速器44の他のバイブライン・ユニットの「ピア」である。即ち、各バイブライン・ユニット78はホストプロセッサ42或は他の任意のバイブライン・ユニットと直接通信できる。よって、このピア-ベクトル・アーキテクチャは、もしバイブライン・ユニット78の全てがマスターバイブライン・ユニット(不図示)或はホストプロセッサ42等の中央箇所を通じて通信した場合に生ずることとなるデータ「ボトルネック」を防止する。更には、これは、マシンに対する重大な変更なしに、ピアを追加するか或はピア-ベクトル・マシン40(図3)からピアを除去することを可能とする。

30

## 【0049】

バイブライン回路80は通信インターフェース82を含み、それが、ホストプロセッサ42(図3)等のピアと、通信シェル84を介したハードウェアに組み込まれたバイブライン74<sub>1</sub>~74<sub>n</sub>(図3)コントローラ86、例外マネージャ88、並びに、コンフィギュレーション・マネージャ90等の、バイブライン回路の他の構成要素との間でデータを転送する。バイブライン回路80は工業規格バス・インターフェース91をも含み得る。代替的には、インターフェース91の機能は通信インターフェース82内に含まれ得る。

## 【0050】

40

バイブライン回路80の複数の構成要素を複数の個別モジュールとして設計することによって、そのバイブライン回路の設計をしばしば簡略化することができる。即ち、それら構成要素の各々を個別に設計及び試験することができ、次いでそれらを統合するものであり、それはソフトウェア或はプロセッサに基づく計算システム(図1のシステム10等)を設計する際に行われることと非常に似ている。加えて、これら構成要素、特に他のバイブライン設計においてたぶん頻繁に使用されるような通信インターフェース82等の構成要素を規定するハードウェア記述言語(HDL)をライブラリ(不図示)内に記憶でき、よって同一構成要素を使用する将来のバイブライン設計の設計及び試験の時間を低減する。即ち、ライブラリからHDLを使用することによって、設計者はスタッフから先行して具現化された構成要素を再設計する必要性がなく、よって設計者の努力を先行して具現化されていない構成要素の設計に対し、或は、先行して具現化された構成要素の変更に對して集中できる。更には、ライブラリ内にバイブライン回路80或はバイブライン加速

50

器 4 4 の多数バージョンを両成する H D L を記憶できて、既存の設計の中から精選及び選択できるように為す。

【0051】

通信インターフェース 8 2 はメッセージ・ハンドラー 6 4 (図 3) によって認識されるフォーマットでデータを送受信し、よってピア・ベクトル・マシン 4 0 (図 3) の設計及び変更を典型的には補助する。例えば、もしデータ・フォーマットが高速 1 / 0 フォーマット等の工業規格であれば、ホストプロセッサ 4 2 及び加速器 4 4 の間にカスタムインターフェースを設計する必要がない。更には、パイプライン回路 8 0 に非バス・インターフェースの代わりにパイプライン・バス 5 0 を介してホストプロセッサ 4 2 (図 3) 等の他のピアと通信させることを可能にすることによって、パイプライン・ユニットが追加成は除去されるたびにゼロから非バス・インターフェースを再設計する代わりに、パイプライン・バスにそれらパイプライン・ユニット (又はそれらを保持する回路カード) を單に接続成は接続解除することによってパイプライン・ユニット 7 8 の数を変更できる。

【0052】

ハードウェアに組み込まれたパイプライン 7 4<sub>1</sub> - 7 4<sub>n</sub> は、図 3 と連携されて先に議論されると共に、先行して引用された「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された特許文献 2 に議論されているようにデータに対して各演算を実行し、通信シェル 8 4 はそれらパイプラインをパイプライン回路 8 0 の他の構成要素や該パイプライン回路外部の回路 (以下に議論されるデータ・メモリ 9 2 等) とインターフェースする。

【0053】

コントローラ 8 6 はハードウェアに組み込まれたパイプライン 7 4<sub>1</sub> - 7 4<sub>n</sub> を同期し、通信に応じて、即ち他のピアからの「事象」に応じて、それらが各データ演算を実行するシーケンスをモニタし且つ制御する。例えば、ホストプロセッサ 4 2 等のピアはパイプライン・バス 5 0 を介してパイプライン・ユニット 7 8 に事象を送信し得て、そのピアがデータから成るブロックをパイプライン・ユニットに送信し終えたことを示して、ハードウェアに組み込まれたパイプライン 7 4<sub>1</sub> - 7 4<sub>n</sub> にそのデータの処理を始めさせる。データを含む事象は典型的にはメッセージと呼称され、データを含まない事象は典型的には「ドアベル」と呼称される。更には、図 5 と連携して以下に議論されるように、パイプライン・ユニット 7 8 も同期信号に応じてパイプライン 7 4<sub>1</sub> - 7 4<sub>n</sub> を同期し得る。

【0054】

例外マネージャ 8 8 はハードウェアに組み込まれたパイプライン 7 4<sub>1</sub> - 7 4<sub>n</sub>、通信インターフェース 8 2、通信シェル 8 4、コントローラ 8 6、並びに、バス・インターフェース 9 1 の状況をモニタし、ホストプロセッサ 4 2 (図 3) に例外を報告する。例えば、もし通信インターフェース 8 2 におけるバッファがオーバーフローすれば、例外マネージャ 8 8 はこれをホストプロセッサ 4 2 に報告する。例外マネージャはその例外を生んだ問題を修正するか成はその修正を試みることも可能である。例えば、オーバーフローしているバッファに対して例外マネージャ 8 8 は、直接的成は以下に議論されるようなコンフィギュレーション・マネージャ 9 0 を介して、そのバッファのサイズを増大し得る。

【0055】

コンフィギュレーション・マネージャ 9 0 はハードウェアに組み込まれたパイプライン 7 4<sub>1</sub> - 7 4<sub>n</sub>、通信インターフェース 8 2、通信シェル 8 4、コントローラ 8 6、例外マネージャ 8 8、並びに、インターフェース 9 1 のソフト・コンフィギュレーションを、ホストプロセッサ 4 2 (図 3) からのソフト・コンフィギュレーション・データに応じて設定し、これは先に引用された「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された特許文献 2 に議論され、ハード・コンフィギュレーションはパイプライン回路 8 0 のトランジスタ及び回路ブロックのレベル上における実際のトポロジーを示し、ソフト・コンフィギュレーションはハード構成された構成要素の物理的パラメータ (例えば、データ幅、テーブル・サイズ) を示す。即ち、ソフト・コンフィギュレーション・データはプロセッサ (図 4 に不図示) のレジスタにロードされ得るプロセッサの動作モード

10

20

30

40

50

ド（例えば、バースト・メモリ・モード）を設定するデータと同様である。例えばホストプロセッサ42は、コンフィギュレーション・マネージャ90に通信インターフェース82におけるキューの数及び各優先レベルを設定させるソフト・コンフィギュレーション・データを送信し得る。例外マネージャ88は、コンフィギュレーション・マネージャ90に、例えば、通信インターフェース82におけるオーバーフローしているバッファのサイズを増大させるソフト・コンフィギュレーション・データをも送信し得る。

## 【0056】

更に図4で参照されるように、バイブライン回路80に加えて、加速器44のバイブライン・ユニット78は、データ・メモリ92、任意選択的な通信バス94、並びに、バイブライン回路がP.L.I.Cである場合におけるファームウェア・メモリ52（図3）を有する。

## 【0057】

データ・メモリ92は、ホストプロセッサ42（図3）等の別のビアとハードウェアに組み込まれたバイブライン74<sub>1</sub>～74<sub>4</sub>との間を流れているデータをバッファすると共に、ハードウェアに組み込まれたバイブラインに対する作業メモリでもある。通信インターフェース82はデータ・メモリ92をバイブライン・バス50（通信バス94ともしあれば工業規格インターフェース91とを介して）にインターフェースし、通信シェル84はデータ・メモリをハードウェアに組み込まれたバイブライン74<sub>1</sub>～74<sub>4</sub>にインターフェースする。

## 【0058】

工業規格バス・インターフェース91は、通信インターフェース82から幾つかのインターフェース回路を効率的にオフロードすることによって、通信インターフェース82のサイズ及び複雑性を低減する従来のバス・インターフェース回路である。それ故に、もしバイブライン・バス50或はルータ61（図3）のパラメータを変更することを望めば、インターフェース91を変更するだけでなく、通信インターフェース82を変更する必要がない。代替的には、バイブライン回路80の外部であるI.C（不図示）内にインターフェース91を配置し得る。バイブライン回路80からインターフェース91をオフロードすることは、例えばハードウェアに組み込まれたバイブライン74<sub>1</sub>～74<sub>4</sub>及びコントローラ86の用途のバイブライン回路上のリソースを解放する。或は、先に議論されたように、バス・インターフェース91は通信インターフェース82の一部であり得る。

## 【0059】

バイブライン回路80がP.L.I.Cである図3と連携されて先に議論されたように、ファームウェア・メモリ52はバイブライン回路のハード・コンフィギュレーションを設定するファームウェアを記憶する。このメモリ52はファームウェアを加速器44の構成中にバイブライン回路80にロードし、加速器の構成中或はその後に通信インターフェース82を介してホストプロセッサ42（図3）から変更されたファームウェアを受信し得る。ファームウェアのローディング及び受信は、先行して引用された「プログラマブル回路、関連計算マシン、並びに、方法」と題された特許文献4に更に議論されている。

## 【0060】

図4で更に参照されるように、バイブライン回路80、データ・メモリ92、並びに、ファームウェア・メモリ52は回路ボード或はカード98上に配置され得て、パソコンナルコンピュータ（不図示）におけるドーターカードがマザーボードのスロットにプラグインされ得ることと非常に類似して、バイブライン・バス・コネクタ（不図示）にプラグインされ得る。図示されていないが、従来のI.C、電力調整器等の構成要素、並びに、電力シーケンサも同様のようにカード98上に配置され得る。

## 【0061】

バイブライン・ユニット78の構造及び動作の更なる詳細は図5と連携されて以下に議論される。

## 【0062】

図5は、本発明の実施例に従った図4のバイブライン・ユニット78のブロック線図で

ある。簡略化のため、ファームウェア・メモリ 52 は図 5 から省略されている。バイブライン回路 80 はマスター CLOCK 信号を受信し、それが直接的或は間接的にバイブルайн回路の以下に記載される構成要素を駆動する。バイブルайн回路 80 は従来方式でマスター CLOCK 信号から 1 つ或はそれ以上のスレーブ CLOCK 信号(不図示)を生成し得る。またバイブルайн回路 80 は以下に議論されるように同期信号 SYNC を受信し得る。

## 【0063】

データ・メモリ 92 は入力デュアル-ポート-スタティック-ランダム-アクセス・メモリ (DPSRAM) 100、出力 DPSRAM 102、並びに、任意選択的な作業 DPSRAM 104 を含む。

10

## 【0064】

入力 DPSRAM 100 は、通信インターフェース 82 を介して、ホストプロセッサ 42 (図 3) 等のピアからデータを受信するための入力ポート 106 を含むと共に、通信シェル 84 を介してそのデータをハードウェアに組み込まれたバイブルайн 74<sub>1</sub>、-74<sub>2</sub> に提供するための出力ポート 108 を含む。一方がデータ入力のためであり且つ他方がデータ出力のためであるこれら 2 つのポートを有することは、DPSRAM 100 へのデータ転送 / DPSRAM 100 からのデータ転送の速度及び効率を増大するが、その理由としては通信インターフェース 82 が DPSRAM にデータを書き込むことができる一方でバイブルайн 74<sub>1</sub>、-74<sub>2</sub> がその DPSRAM からデータを読み取ることができるからである。更に、先に議論されたように、ホストプロセッサ 42 等のピアからデータをバッファすべく DPSRAM 100 を用いることは、そのピア及びバイブルайн 74<sub>1</sub>、-74<sub>2</sub> に相互に対しても非同期的に動作させることを可能としている。即ちピアは、バイブルайн 74<sub>1</sub>、-74<sub>2</sub> が現行動作を完了するのを「待機」することなしに、データをバイブルайнに送信できる。同様に、バイブルайн 74<sub>1</sub>、-74<sub>2</sub> はピアがデータ送信動作を完了するのを「待機」することなしにデータを検索できる。

20

## 【0065】

同じように、出力 DPSRAM 102 は、通信シェル 84 を介して、ハードウェアに組み込まれたバイブルайн 74<sub>1</sub>、-74<sub>2</sub> からデータを受信するための入力ポート 110 を含むと共に、そのデータを通信インターフェース 82 を介してホストプロセッサ 42 (図 3) 等のピアに提供するための出力ポート 112 を含む。先に議論されたように、これら 2 つのデータ・ポート 110 (入力) 及び 112 (出力) は DPSRAM 102 へのデータ転送 / DPSRAM 102 からのデータ転送の速度及び効率を増大し、バイブルайн 74<sub>1</sub>、-74<sub>2</sub> からデータをバッファすべく DPSRAM 102 を用いることは、ピア及びバイブルайнに相互に対しても非同期的に動作させることを可能としている。即ち、バイブルайн 74<sub>1</sub>、-74<sub>2</sub> は、出力データ・ハンドラー 126 がそのピア或は別のピアにデータ転送を完了するのを「待機」することなしにピアにデータを発行できる。同様に、出力データ・ハンドラー 126 は、バイブルайн 74<sub>1</sub>、-74<sub>2</sub> がデータ発行動作を完了するのを「待機」することなしにデータをピアに転送できる。

30

## 【0066】

作業 DPSRAM 104 は、通信シェル 84 を介して、ハードウェアに組み込まれたバイブルайн 74<sub>1</sub>、-74<sub>2</sub> からデータを受信するための入力ポート 114 を含むと共に、その通信シェルを介してそのデータをバイブルайнに渡すための出力ポート 116 を含む。DPSRAM 100 から受信された入力データを処理している間に、バイブルайн 74<sub>1</sub>、-74<sub>2</sub> は部分的に処理された、即ち中間データをそのデータの処理を続行する前に一時的に記憶する必要があり得る。例えば、バイブルайн 74<sub>1</sub> 等の第 1 バイブルайнはバイブルайн 74<sub>2</sub> 等の第 2 のバイブルайнによる更なる処理のために中間データを生成し得て、よって、第 1 バイブルайнは第 2 バイブルайнがその中間データを検索するまでそれを一時的に記憶する必要があり得る。作業 DPSRAM 104 はこの一時的記憶装置を提供する。先に議論されたように、2 つのデータ・ポート 114 (入力) 及び 116 (出力) はバイブルайн 74<sub>1</sub>、-74<sub>2</sub> 及び DPSRAM 104 の間でのデータ転送の速度及び効率を

40

50

率を増大する。更には別個の作業 D P S R A M 1 0 4 を含むことは、D P S R A M 1 0 0 及び 1 0 2 にデータ入力バッファ及びデータ出力バッファ専用としてそれぞれ機能させることによってパイプライン回路 8 0 の速度及び効率を典型的には増大する。しかし、パイプライン回路 8 0 に対する僅かな変更によって、D P S R A M 1 0 0 及び 1 0 2 の両方或は片方が、D P S R A M 1 0 4 が省略された際に、そしてそれが存在したとしても、パイプライン 7 4<sub>1</sub> - 7 4<sub>2</sub> に対する作業メモリでもあり得る。

## 【0067】

D P S R A M 1 0 0, 1 0 2, 1 0 4 はパイプライン回路 8 0 の外部であるとして記載されているが、それら D P S R A M 若しくはそれと同等物の内の 1 つ或はそれ以上をそのパイプライン回路の内部とすることが可能である。

10

## 【0068】

更に図 5 で参照されるように、通信インターフェース 8 2 は、工業規格バス・アダプタ 1 1 8、入力データ・ハンドラー 1 2 0、入力データ及び入力事象キュー 1 2 2 及び 1 2 4、出力データ・ハンドラー 1 2 6、並びに、出力データ及び出力事象キュー 1 2 8 及び 1 3 0 を含む。それらのキュー 1 2 2, 1 2 4, 1 2 8, 1 3 0 がそれぞれ单一キューとして示されているが、それらキューの内の 1 つ或はそれ以上は、例えば、それらキューに記憶された値の優先順位、或は、それら値が表す各データの優先順位による分離を可能とするサブ・キュー（不図示）を含み得る。

## 【0069】

工業規格バス・アダプタ 1 1 8 は、通信バス 9 4 を介して、パイプライン回路 8 0 及びパイプライン・バス 5 0（図 4）の間でのデータの転送を可能とする物理層を含む。それ故に、もしバス 9 4 のパラメータの変更が望まれたならば、アダプタ 1 1 8 だけを変更する必要があり、通信インターフェース 8 2 全体を変更する必要性はない。工業規格バス・インターフェース 9 1 がパイプライン・ユニット 7 8 から省略されている場合、アダプタ 1 1 8 をパイプライン・バス 5 0 及びパイプライン回路 8 0 の間での直接的なデータ転送を可能とするよう変更し得る。この後者の具現化例において、変更アダプタ 1 1 8 はバス・インターフェース 9 1 の機能を含み、バス 5 0 のパラメータの変更が望まれたならばアダプタ 1 1 8 を変更することだけが必要である。

20

## 【0070】

入力データ・ハンドラー 1 2 0 は工業規格アダプタ 1 1 8 からデータを受信し、そのデータを入力ポート 1 0 6 を介して D P S R A M 1 0 0 にロードし、そのデータを指すポイントと対応するデータ識別子を生成してそれらを入力データ・キュー 1 2 2 に記憶する。もしそのデータがホストプロセッサ 4 2（図 3）等のピアからのメッセージのペイロードであれば、入力データ・ハンドラー 1 2 0 はそのデータを D P S R A M 1 0 0 にロードする前にメッセージからそれを抽出する。入力データ・ハンドラー 1 2 0 はインターフェース 1 3 2 を含み、それが D P S R A M 1 0 0 の入力ポート 1 0 6 にデータを書き込むが、それは図 6 と連携された以下に更に細織される。代替的に、入力データ・ハンドラー 1 2 0 は抽出ステップを省略できると共にメッセージ全体を D P S R A M 1 0 0 にロードできる。

30

## 【0071】

また入力データ・ハンドラー 1 2 0 は工業規格アダプタ 1 1 8 から事象を受信して、その事象を入力事象キュー 1 2 4 にロードする。

40

## 【0072】

更には、入力データ・ハンドラー 1 2 0 は有効マネージャ 1 3 4 を含み、それが受信されたデータ或は事象がパイプライン回路 8 0 に対して意図されているかを決定する。有効マネージャ 1 3 4 は、データ或は事象を含むメッセージのヘッダー（或はその一部）を分析することによって、データ或は事象のタイプを分析することによって、或は、データ或は事象の例証識別（即ちデータ/事象が意図されているハードウェアに組み込まれたパイプライン 7 4）を分析することによって、先の決定を為し得る。もし入力データ・ハンドラー 1 2 0 がパイプライン回路 8 0 に対して意図されていないデータ或は事象を受信すれ

50

ば、有効マネージャ 134 は入力データ・ハンドラーがその受信データ／事象をロードすることを禁止する。ピア-ベクトル・マシン 40 がルータ 61 (図 3) を含んで、パイプライン・ユニット 78 が該パイプライン・ユニットに対して意図されているデータ／事象のみを受信するように為す場合、有効マネージャ 134 も入力データ・ハンドラー 120 に例外 (誤って受信されたデータ／事象) やその例外を引き起こしたピアを識別する例外メッセージをホストプロセッサ 42 (図 3) に送信させ得る。

## 【0073】

出力データ・ハンドラー 126 は出力データ・キュー 128 によって指された D P S R A M 102 の複数箇所から処理データを検索し、それら処理データを工業規格バス・アダプタ 118 を介してホストプロセッサ 42 (図 3) 等の 1 つ或はそれ以上のピアに送信する。出力データ・ハンドラー 126 はインターフェース 136 を含み、それがポート 112 を介して D P S R A M 102 から処理データを読み取る。このインターフェース 136 は圖 7 と連携されて以下に更に議論される。

## 【0074】

出力データ・ハンドラー 126 もパイプライン 741-748 によって生成された事象を出力事象キュー 130 から検索し、工業規格バス・アダプタ 118 を介してホストプロセッサ 42 (図 3) 等の 1 つ或はそれ以上のピアにその検索された事象を送信する。

## 【0075】

更には、出力データ・ハンドラー 126 は加入マネージャ 138 を含み、それが処理データや事象に加入しているホストプロセッサ 42 (図 3) 等のピアのリストを含み、出力データ・ハンドラーはそのリストを用いてデータ／事象を正しいピアに送信する。もしピアがデータ／事象がメッセージのペイロードであることを好めば、出力データ・ハンドラー 126 は加入マネージャ 138 からピアのネットワーク或はバス-ポート・アドレスを検索し、アドレスを含むヘッダーを生成し、データ／事象及びヘッダーからメッセージを生成する。

## 【0076】

D P S R A M 100 及び 102 にデータを記憶し記憶されたデータを検索するための技術はポインタやデータ識別子の使用を含むが、他のデータ管理技術を具現化すべく入力及び出力データ・ハンドラー 120 及び 126 を変更することができる。こうしたデータ管理技術の従来例は、キー或はトークンを用いるポインタ、入力／出力制御 (I O C) ブロック、並びに、スプリーリングを含む。

## 【0077】

通信シェル 84 はハードウェアに組み込まれたパイプライン 741-748 を出力データ・キュー 128、コントローラ 86、並びに、D P S R A M 100、102、104 にインターフェースする物理層を含む。このシェル 84 はインターフェース 140 及び 142 と任意選択的なインターフェース 144 及び 146 を含む。インターフェース 140 及び 146 はインターフェース 136 と同様であり得て、インターフェース 140 はポート 108 を介して D P S R A M 100 から入力データを読み取り、そしてインターフェース 146 はポート 116 を介して D P S R A M 104 から中間データを読み取る。インターフェース 142 及び 144 はインターフェース 132 と同様であり得て、インターフェース 142 はポート 110 を介して D P S R A M 102 に処理データを書き込み、そしてインターフェース 144 はポート 114 を介して D P S R A M 104 に中間データを書き込む。

## 【0078】

コントローラ 86 はシーケンス・マネージャ 148 及び同期インターフェース 150 を含み、該同期インターフェースは 1 つ或はそれ以上の同期信号 S Y N C を受信する。ホストプロセッサ 42 (図 3) 等のピア、或は、ピア-ベクトル・マシン 40 (図 3) の外部の装置 (不図示) は、S Y N C 信号を生成し得て、以下に議論されると共に、先行して引用された「多数パイプライン・ユニットを有するパイプライン加速器、関連計算マシン、並びに、方法」と題された特許文献 5 に議論されるように、それがシーケンス・マネージ

10

20

30

40

50

ヤ148をトリガーしてハードウェアに組み込まれたバイブライン74<sub>1</sub>～74<sub>6</sub>を起動する。同期インターフェース150もSYNC信号を生成し得て、バイブライン回路80をトリガーするか或は別のピアをトリガーする。加えて、入力事象キュー124からの事象もシーケンス・マネージャ148をトリガーして、以下に議論されるように、ハードウェアに組み込まれたバイブライン74<sub>1</sub>～74<sub>6</sub>を起動する。

## 【0079】

シーケンス・マネージャ148は通信シェル84を介してハードウェアに組み込まれたバイブライン74<sub>1</sub>～74<sub>6</sub>をそれらの各動作を通じて順序付けする。典型的には、各バイブライン74<sub>i</sub>は、事前処理、処理、並びに、事後処理の少なくとも3つの状態を有する。事前処理中、バイブライン74<sub>i</sub>は、例えば、そのレジスタを初期化して、DPSRAM100から入力データを検索する。処理中、バイブライン74<sub>i</sub>は、例えば、その検索されたデータに対して演算を為し、中間データをDPSRAM104に一時的に記憶し、DPSRAM104からその中間データを検索してから、その中間データに対して演算を為して結果データを生成する。事後処理中、バイブライン74<sub>i</sub>は、例えば、その結果としてのデータをDPSRAM102にロードする。それ故に、シーケンス・マネージャ148はバイブライン74<sub>1</sub>～74<sub>6</sub>の動作又は演算をモニタして、各バイブラインにその動作状態の各々をいつ始めるべきかを命令する。そして、バイブライン・タスクを先に記載したものとは異なるように各種動作状態の間に分配し得る。例えば、バイブライン74<sub>i</sub>は事前処理状態中の代わりに処理状態中にDPSRAM100から入力データを検索し得る。

## 【0080】

更には、シーケンス・マネージャ148はハードウェアに組み込まれたバイブライン74<sub>1</sub>～74<sub>6</sub>間で所定の内部動作同期を維持する。例えば、バイブライン74<sub>1</sub>～74<sub>6</sub>の全てがDPSRAM100からデータを同時に検索することを回避すべく、第1バイブライン74<sub>1</sub>が事前処理状態である一方で、第2バイブライン74<sub>2</sub>が処理状態、第3バイブライン74<sub>3</sub>が事後処理状態となるようにそれらバイブラインを同期させることができ、一つのバイブライン74<sub>i</sub>の状態が別のバイブラインの同時発生的に実行されている状態とは異なる数のクロック・サイクルを要求し得るので、バイブライン74<sub>1</sub>～74<sub>6</sub>はもし自由に走らされると同期性を喪失し得る。結果として、特定時に、例えば多数のバイブライン74<sub>i</sub>がDPSRAM100からデータを同時に検索しようとするような「ボトルネック」があり得る。同時性の喪失やその望ましくない結果を防止すべく、シーケンス・マネージャ148は、それらバイブラインの任意のものに対して次の動作状態に進める前に、バイブライン74<sub>i</sub>の全てに現行の動作状態を完了させる。それ故に、シーケンス・マネージャ148が現行の動作状態に対して振り分ける時間は、最も緩慢なバイブライン74<sub>i</sub>にその状態を完了させるに充分な長さである。代替的には、ハードウェアに組み込まれたバイブライン74<sub>1</sub>～74<sub>6</sub>の間で所定動作同期性を維持するための回路（不図示）はバイブライン自体内に含まれ得る。

## 【0081】

ハードウェアに組み込まれたバイブライン74<sub>1</sub>～74<sub>6</sub>を順序づけすると共に内部的に同期化することに加えて、シーケンス・マネージャ148はバイブラインの動作を、1つ或はそれ以上のSYNC信号、或は、入力事象キュー124内の事象に応じて、ホストプロセッサ42（図3）等の他のピアの動作、及び他の外部装置の動作と同期する。

## 【0082】

典型的には、SYNC信号はタイムクリティカルな機能をトリガーするが若しいハードウェア・リソースを必要とし、比較して、典型的には事象は非タイムクリティカルな機能をトリガーするが若しくより少ないハードウェア・リソースを必要とする。先行して引用された「多数バイブライン・ユニットを有するバイブライン加速器、関連計算マシン、並びに、方法」と題された特許文5に議論されているように、SYNC信号はピアからピアに直接的に経路指定されているので、例えば、バイブライン・バス50（図3）、入力データ・ハンドラー120、並びに、入力事象キュー124を通じてその道を作成しなければならない事象よりも迅速に機能をトリガーできる。しかし、それらが個別に経路指定

10

20

30

40

50

されるので、SYNC信号は、ルーティング・ライン、バッファ、並びに、SYNCインターフェース150等のバイブライン回路80の専用回路を必要とする。逆に、それらが既存のデータ転送下部組織（例えば、バイブルайн・バス50及び入力データ・ハンドラー120）を使用するので、事象は専用の入力事象キュー124のみを必要とする。結果として、設計者はタイムクリティカルな機能のほとんど全てをトリガーすべく事象を使用する傾向がある。

#### 【0083】

以下は機能トリガーリングの一例である。ソナー・センサ要素（不図示）がバイブルайн・ユニット78にデータから成るブロックを送信することを仮定すると、入力データ・ハンドラー120はこのデータをDPSRAM100に記憶し、バイブルайн74<sub>1</sub>はこのデータをそのDPSRAM100からDPSRAM104に転送し、そして、トリガーされると、バイブルайн74<sub>2</sub>はDPSRAM104からそのデータを検索して処理する。もしバイブルайн74<sub>2</sub>がそのデータに実行する処理がタイムクリティカルであれば、バイブルайн74<sub>1</sub>がDPSRAM104にデータ・ブロックの全体をローディングし終えるとすぐに、センサ要素はSYNCパルスを生成してバイブルайн74<sub>2</sub>をインターフェース150及びシーケンス・マネージャ148を介してトリガーする。バイブルайн・ユニット78及びセンサがバイブルайн74<sub>1</sub>がいつ終了されるかを決定すべく利用できる数多くの従来技術がある。例えば、以下に議論されるように、シーケンス・マネージャ148は対応するSYNCパルス或は事象をセンサに提供し得る。代替的には、もしバイブルайн74<sub>2</sub>が実行する処理がタイムクリティカルでなければ、センサはバイブルайн・バス50（図3）を介して事象をシーケンス・マネージャ148に送信し得る。

#### 【0084】

シーケンス・マネージャ148もホストプロセッサ42（図3）等のピアにSYNCパルス或は事象を生成することによってハードウェアに組み込まれたバイブルайн74<sub>1</sub>・74<sub>2</sub>の動作に関する情報を提供し得る。シーケンス・マネージャ148はSYNCインターフェース150及び専用ライン（不図示）を介してSYNCパルスを送信し、出力事象キュー130及び出力データ・ハンドラー126を介して事象を送信する。先の例で参照されるように、ピアがバイブルайн74<sub>2</sub>からのデータ・ブロックを更に処理すると仮定する。シーケンス・マネージャ148は、SYNCパルス或は事象を介して、バイブルайн74<sub>2</sub>がデータから成るブロックをいつ処理し終えたかをそのピアに通知する。シーケンス・マネージャ148も、対応するSYNCパルス或は事象を生成して適切なピア（単数或は複数）に送信することによって、SYNCパルス或は事象の受領を確認し得る。

#### 【0085】

更に図5で参照されるようにバイブルайн・ユニット78の動作は本発明の実施例に従って議論される。

#### 【0086】

データに対して、工業規格バス・インターフェース91はバイブルайн・バス50（及び、もし有ればルータ61）からデータ信号（ホストプロセッサ42（図3）等のピアに起因する）を受信し、それら信号をヘッダー及びペイロードを各々が有するメッセージに変換する。

#### 【0087】

次に、工業規格バス・アダプタ118はその工業規格バス・インターフェース91からのメッセージを入力データ・ハンドラー120と互換性があるフォーマットに変換する。

#### 【0088】

次いで、入力データ・ハンドラー120はそのメッセージ・ヘッダを細かく調べて、各ヘッダーからデータ・ペイロードを記述する部分を抽出する。例えば、抽出されたヘッダ一部分は、例えば、バイブルайн・ユニット78のアドレス、ペイロード中のデータのタイプ、或は、そのデータが意図されているバイブルайн78<sub>1</sub>・78<sub>2</sub>を識別する例証識別子を含み得る。

#### 【0089】

10

20

30

40

50

次に、有効マネージャ 134 はその抽出ヘッダー部分を分析し、そのデータがハードウェアに組み込まれたバイ二進法 741-740 の内の 1 つに意図されていることを確認し、インターフェース 132 はそのデータをポート 106 を介して D P S R A M 100 の箇所に書き込み、そして、入力データ・ハンドラー 120 はその箇所と対応するデータ識別子に対するポインタを入力データ・キュー 122 に記憶する。データ識別子はデータが意図されているバイ二進法或は複数のバイ二進法 741-740 を識別するか、或は、シーケンス・マネージャ 148 に以下に議論されるようにその識別を行わせることを可能とする情報を含む。代替的には、キュー 122 は各バイ二進法 741-740 に対する各サブキュー（不図示）を含み得て、入力データ・ハンドラー 120 は意図されたバイ二進法或は意図された複数のバイ二進法のサブキュー或は複数のサブキュー内にポインタを記憶する。この代替例において、データ識別子は省略し得る。更には、もしデータがメッセージのペイロードであれば、入力データ・ハンドラー 120 はインターフェース 132 が D P S R A M 100 内にそのデータを記憶する前にメッセージからそのデータを抽出する。代替的には、先に議論されたように、インターフェース 132 はメッセージ全体を D P S R A M 100 内に記憶し得る。

10

## 【0090】

次いで、適時に、シーケンス・マネージャ 148 は入力データ・キュー 122 からポインタ及びデータ識別子を読み取って、そのデータ識別子から、データが意図されているバイ二進法或は複数のバイ二進法 741-740 を決定し、そのポインタを通信シェル 84 を介してそのバイ二進法或は複数のバイ二進法に渡す。

20

## 【0091】

次に、データ受信バイ二進法或は複数のデータ受信バイ二進法 741-740 はインターフェース 140 にポート 108 を介して D P S R A M 100 の指された箇所からデータを検索させる。

20

## 【0092】

次いで、データ受信バイ二進法或は複数のデータ受信バイ二進法 741-740 は検索されたデータを処理し、インターフェース 142 はその処理されたデータをポート 110 を介して D P S R A M 102 のある箇所に書き込み、そして、通信シェル 84 は出力データ・キュー 128 にその処理されたデータを指すポインタとその処理されたデータに対するデータ識別子とをロードする。そのデータ識別子は、その処理データに加入しているホストプロセッサ 42（図 3）等の仕向先ビア或は複数の仕向先ビアを識別するか、加入マネージャ 138 に引き続き仕向先ビア或は複数の仕向先ビア（例えば、図 3 のホストプロセッサ 42）を決定させることを可能とする情報を（データ・タイプ等）を含む。代替的には、キュー 128 は各バイ二進法 741-740 に対する各サブキュー（不図示）を含み得て、通信シェル 84 は起因バイ二進法或は複数の起因バイ二進法のサブキュー或は複数のサブキューにポインタを記憶する。この代替例において、通信シェル 84 はデータ識別子のキュー 128 へのローディングを省略し得る。更には、もしバイ二進法或は複数のバイ二進法 741-740 が検索されたデータを処理している間に中間データを生成すれば、インターフェース 144 はその中間データをポート 114 を介して D P S R A M 104 に書き込み、インターフェース 146 はポート 116 を介してその D P S R A M 104 から中間データを検索する。

30

## 【0093】

次に、出力データ・ハンドラー 126 は出力データ・キュー 128 からポインタ及びデータ識別子を検索し、加入マネージャ 138 はその識別子からデータの仕向先ビア或は複数の仕向先ビア（たとえば、図 3 のホストプロセッサ 42）を決定し、インターフェース 136 はポート 112 を介して D P S R A M 102 の指された箇所からデータを検索し、出力データ・ハンドラー 126 はそのデータを工業規格バス・アダプタ 118 に送信する。もし仕向先ビアがメッセージのペイロードであるデータを必要とすれば、出力データ・ハンドラー 126 はそのメッセージを生成してそのメッセージをアダプタ 118 に送信する。例えば、データが多数の仕向先ビアを有して、バイ二進法・バス 50 がメッセージ放送を

40

50

支援すると仮定する。出力データ・ハンドラー 126 はその仕向先ビア全てのアドレスを含む单一ヘッダーを生成し、そのヘッダー及びデータを 1 つのメッセージとして組み合わせて、單一メッセージを仕向先ビアの全てに（アダプタ 118 及び工業規格バス・インターフェース 91 を介して）同時に送信する。代替的には、出力データ・ハンドラー 126 は各ヘッダー、よって各メッセージを各仕向先ビアに対して生成し、それらメッセージの各々を個別に送信する。

## 【0094】

次いで、工業規格バス・アダプタ 118 は出力データ・ハンドラー 126 からのデータをフォーマットして、工業規格バス・インターフェース 91 と互換性を持たせるように為す。

## 【0095】

次に、工業規格バス・インターフェース 91 は工業規格バス・アダプタ 118 からのデータをフォーマットして、パイプライン・バス 50（図 3）と互換性を持たせるように為す。

## 【0096】

随伴データを伴わない事象、即ちドアベルに対して、工業規格バス・インターフェース 91 はパイプライン・バス 50 から（もしあればルータ 61 からも）信号（図 3 のホストプロセッサ 42 等のビアに起因する）を受信し、その信号を事象を含むヘッダー（即ち、データ無しメッセージ）に変換する。

## 【0097】

次に、工業規格バス・アダプタ 118 は工業規格バス・インターフェース 91 からのヘッダーを入力データ・ハンドラー 120 と互換性があるフォーマットに変換する。

## 【0098】

次いで、入力データ・ハンドラー 120 はそのヘッダーから事象と事象の記述とを抽出する。例えば、記述は、例えは、パイプライン・ユニット 78 のアドレス、事象のタイプ、或は、その事象が意図されているパイプライン 78,-78 を識別する例証識別子を含み得る。

## 【0099】

次に、有効マネージャ 134 は事象記述を分析し、その事象がハードウェアに組み込まれたパイプライン 74,-74 の内の 1 つに意図されていることを確認して、入力データ・ハンドラー 120 はその事象とその記述とを入力事象・キューリスト 124 に記憶する。

## 【0100】

次いで、適時に、シーケンス・マネージャ 148 は入力事象・キューリスト 124 からの事象及びその記述を読み取り、その事象に応じて、先に議論されたように、パイプライン 74,-74 の内の 1 つ或はそれ以上の動作をトリガーする。例えは、シーケンス・マネージャ 148 はパイプライン 74 をトリガーし得て、パイプライン 74 が先行して DPSR AM 104 に記憶したデータの処理を始める。

## 【0101】

事象を出力すべく、シーケンス・マネージャ 148 は事象とその事象の記述を生成し、それら事象及び記述を出力事象・キューリスト 130 にロードする。事象記述は、もし 2 つ以上の可能性ある仕向先ビアが存在する場合にその仕向先ビア（単数或は複数）を識別する。例えは、先に議論されたように、事象は入力事象、入力データ或は入力事象のメッセージ、或は、SYNC パルスの受領及び具現化を確認し得る。

## 【0102】

次に、出力データ・ハンドラー 126 は事象とその記述を出力事象・キューリスト 130 から検索し、加入マネージャ 138 はその事象記述から該事象の仕向先ビア或は複数の仕向先ビア（例えは、図 3 のホストプロセッサ 42）を決定し、出力データ・ハンドラーは、先に議論されたように、その事象を工業規格バス・アダプタ 118 及び工業規格バス・インターフェース 91 を介して適切な仕向先ビア或は複数の適切な仕向先ビアに送信する。

## 【0103】

10

20

30

40

50

コンフィギュレーション・コマンドに対して、工業規格バス・アダプタ118は工業規格バス・インターフェース91を介してホストプロセッサ42(図3)からコマンドを受信し、そのコマンドをデータ無し事象(即ち、ドアベル)に対して先に議論されたものと同様の方式で入力データ・ハンドラー120に提供する。

## 【0104】

次に、有効マネージャ134はそのコマンドがパイプライン・ユニット78に対して意図されていることを確認し、入力データ・ハンドラー120はそのコマンドをコンフィギュレーション・マネージャ90にロードする。更に、入力データ・ハンドラー120或はコンフィギュレーション・マネージャ90の何れかもそのコマンドを出力データ・ハンドラー126に渡すことができ、それがパイプライン・ユニット78がコマンドを受信したことそのコマンドを送信したビア(例えば、図3のホストプロセッサ42)に送信し戻すことによって確認する。この確認技術はしばしば「エコー」と呼称される。

10

## 【0105】

次いで、コンフィギュレーション・マネージャ90はそのコマンドを具現化する。例えば、コマンドはコンフィギュレーション・マネージャ90にデバッグ目的でパイプライン74<sub>1</sub>～74<sub>n</sub>の内の1つをディスエーブルさせ得る。又は、コマンドはホストプロセッサ42(図3)等のビアに出力データ・ハンドラー126を介してコンフィギュレーション・マネージャ90からパイプライン回路80の現行コンフィギュレーションを読み取らせてることができる。加えて、コンフィギュレーション・コマンドを使用し得て、例外マネージャ88によって認識される例外を規定することができる。

20

## 【0106】

例外に対して、パイプライン回路80の入力データ・キュー122等の構成要素は例外マネージャ88に向けて例外をトリガーする。一具現化例において、その構成要素は当該構成要素をモニタして、所定条件或は各種条件から成る組に応じてその例外をトリガーする例外トリガリング・アダプタ(不図示)を含む。例外トリガリング・アダプタは一度で設計され得てから例外を生成するパイプライン回路80の各構成要素の一部として含まれ得る普遍的回路であり得る。

## 【0107】

次に、例外トリガーに応じて、例外マネージャ88は例外識別子を生成する。例えば、その識別子は入力データ・キュー122がオーバーフローしたことを示し得る。更には、その識別子は、もし2つ以上の可能性ある仕向先ビアが存在する場合にその仕向先ビアを含み得る。

30

## 【0108】

次いで、出力データ・ハンドラー126は例外マネージャ88から例外識別子を検索し、その例外識別子を、先行して引用された「改善された計算アーキテクチャを有する計算マシン、関連システム、並びに、方法」と題された特許文献3で議論されたようにホストプロセッサ42(図3)に送信する。代然的には、もし多数の可能性ある仕向先ビアが存在すれば、例外識別子も仕向先情報を含むことができ、それから加入マネージャ138がその識別子の仕向先ビア或は複数の仕向先ビア(例えば、図3のホストプロセッサ42)を決定する。出力データ・ハンドラー126は、次いで、その識別子を、工業規格バス・アダプタ118及び工業規格バス・インターフェース91を介して、仕向先ビア或は複数の仕向先ビアに送信する。

40

## 【0109】

更に図5で参照されるように、パイプライン・ユニット78に対する代替実施例が存在する。例えば、DPSRAMを含むものとして説明されているが、データ・メモリ92はクワッドデータレート(QDR)SRAM等のメモリICの他のタイプを含み得る。

## 【0110】

図6は、本発明の実施例に従った図5のインターフェース142のブロック線図である。図5と連携して先に議論されたように、インターフェース142はハードウェアに組み込まれたパイプライン74<sub>1</sub>～74<sub>n</sub>からDPSRAM102に処理データを書き込む。以

50

下に議論されるように、インターフェース 142 の構造は、データ「ボトルネック」を低減又はなくし、パイプライン回路 80 (図 5) が P L I C である場合、P L I C のローカル及びグローバルのルーティング・リソースを効率的に利用する。

【0111】

インターフェース 142 は書き込みチャネル 150<sub>1</sub>～150<sub>n</sub>を含み、各ハードウェアに組み込まれたパイプライン 74<sub>1</sub>～74<sub>n</sub> (図 5) に対して 1 つずつのチャネルとなっており、コントローラ 152 を含む。図示の簡略化の目的のため、チャネル 150<sub>1</sub>が以下に議論され、他のチャネル 150<sub>2</sub>～150<sub>n</sub>の動作及び構造が別段の説明がない限り同様であることを理解して頂きたい。

【0112】

チャネル 150<sub>1</sub>は書き込みアドレス/データ FIFO 154<sub>1</sub>及びアドレス/データ・レジスタ 156<sub>1</sub>を含む。

【0113】

FIFO 154<sub>1</sub>は、パイプライン 74<sub>1</sub>が DPSRAM 102 に書き込むデータを記憶し、コントローラ 152 がレジスタ 156<sub>1</sub>を介して DPSRAM 102 に実際にデータを書き込むことができるまで、パイプラインがデータを書き込む DPSRAM 102 内の箇所のアドレスを記憶する。それ故に、FIFO 154<sub>1</sub>は、もしコントローラ 152 が先行するデータの書き込みを終了するまでに、パイプライン 74<sub>1</sub>がチャネル 150<sub>1</sub>へのデータ書き込みを「待機」しなければならなかつた場合に生じ得るデータ・ボトルネックを低減又はなくする。

【0114】

FIFO 154<sub>1</sub>はバス 158<sub>1</sub>を介してパイプライン 74<sub>1</sub>からデータを受信し、バス 160<sub>1</sub>を介してデータが書き込まれることになる箇所のアドレスを受信し、そのデータ及びアドレスをバス 162<sub>1</sub>及び 164<sub>1</sub>を介してレジスタ 156<sub>1</sub>に提供する。更には、FIFO 154<sub>1</sub>はパイプライン 74<sub>1</sub>からライン 166<sub>1</sub>上の書き込み FIFO 信号 (WRITE FIFO 信号) を受信し、ライン 168<sub>1</sub>を介してクロック信号 (CLOCK 信号) を受信し、そして FIFO 充満信号をライン 170<sub>1</sub>上でパイプライン 74<sub>1</sub>に提供する。加えて、FIFO 154<sub>1</sub>はライン 172<sub>1</sub>を介してコントローラ 152 から読み取り FIFO 信号を受信して、ライン 174<sub>1</sub>を介して FIFO EMPTY 信号 (FIFO 空信号) をコントローラに提供する。パイプライン回路 80 (図 5) が P L I C である場合、バス 158<sub>1</sub>、160<sub>1</sub>、162<sub>1</sub>、164<sub>1</sub>及びライン 166<sub>1</sub>、168<sub>1</sub>、170<sub>1</sub>、172<sub>1</sub>、174<sub>1</sub>は、好ましくはローカル・ルーティング・リソースを用いて形成される。典型的には、ローカル・ルーティング・リソースは、信号経路長が一般により短く且つルーティングの具現化がより容易であるため、グローバル・ルーティング・リソースよりも好まれる。

【0115】

レジスタ 156<sub>1</sub>は、バス 162<sub>1</sub>及び 164<sub>1</sub>を介して、FIFO 154<sub>1</sub>から書き込まれるべきデータと、書き込み箇所のアドレスとをそれぞれ受信し、それらデータ及びアドレスをアドレス/データ・バス 176<sub>1</sub>を介して DPSRAM 102 (図 5) のポート 110 に提供する。更には、レジスタ 156<sub>1</sub>もデータ及びアドレスを、以下に議論されるように、アドレス/データ・バス 178<sub>1</sub>を介してレジスタ 156<sub>2</sub>～156<sub>n</sub>から受信する。加えて、レジスタ 156<sub>1</sub>はライン 180<sub>1</sub>を介してコントローラ 152 からシフト/ロード信号 (SHIFT/LOAD 信号) を受信する。パイプライン回路 80 (図 5) が P L I C である場合、バス 176<sub>1</sub>は典型的にはグローバル・ルーティング・リソースを用いて形成され、バス 178<sub>1</sub>～178<sub>n</sub>及びライン 180<sub>1</sub>は好ましくはローカル・ルーティング・リソースを用いて形成される。

【0116】

FIFO 空信号の受信と読み取り FIFO 及びシフト/ロード信号の生成に加えて、コントローラ 152 は書き込み DPSRAM 信号 (WRITE DPSRAM 信号) をライン 182<sub>1</sub>を介して DPSRAM 102 (図 5) のポート 110 に提供する。

10

20

30

40

50

## 【0117】

更に図6で参照されるように、インターフェース142の動作が議論される。

## 【0118】

先ず、FIFO154はFIFOの現行状態（「充満」或は「非充満」）と対応する論理レベルまでFIFO充満信号を駆動する。

## 【0119】

次に、もしFIFO154が充満していなく且つバイブルイン74が書き込むべきデータを処理していれば、バイブルインはデータ及び対応するアドレスをバス158及び160までそれぞれ駆動し、書き込み信号をアサートし、よってデータ及びアドレスをFIFOにロードする。しかしながら、もしFIFO154が充満していれば、バイブルイン74はデータをロードする前にFIFOが非充満となるまで待機する。  
10

## 【0120】

次いで、FIFO154はFIFO空信号をFIFOの現行状態（「空」或は「非空」）と対応する論理レベルまで駆動する。

## 【0121】

次に、もしFIFO154が非空であれば、コントローラ152は読み取りFIFO信号をアサートし、シフト/ロード信号をロード論理レベルまで駆動し、よってFIFOから最初にロードされたデータ及びアドレスをレジスタ156にロードする。もしFIFO154が空であれば、コントローラ152は読み取りFIFOをアサートしないが、他のFIFO154-154の内の任意のものが空でなければ、シフト・ロードをロード論理レベルまで駆動する。  
20

## 【0122】

チャネル150-150は同様の方式で動作して、FIFO154-154に最初にロードされたデータがそれぞれレジスタ156-156にロードされるようになる。

## 【0123】

次いで、コントローラ152はシフト/ロード信号をシフト論理レベルまで駆動し、書き込みDPSRAM信号をアサートし、よってレジスタ156-156からアドレス/データ・バス176にデータ及びアドレスを順次シフトし、DPSRAM102の対応する箇所にデータをロードする。詳細には、第1シフト・サイクル中、レジスタ156からのデータ及びアドレスはバス176にシフトされて、FIFO154からのデータはDPSRAM102のアドレス指定された箇所にロードされるようになる。また第1シフト・サイクル中、レジスタ156からのデータ及びアドレスはレジスタ156にシフトされて、レジスタ156（不図示）からのデータ及びアドレスはレジスタ156にシフトされる、等々である。第2シフト・サイクル中、レジスタ156からのデータ及びアドレスはバス176にシフトされて、FIFO154からのデータはDPSRAM102のアドレス指定された箇所にロードされるようになる。また第2シフト・サイクル中、レジスタ156からのデータ及びアドレスはレジスタ156にシフトされ、レジスタ156（不図示）からのデータ及びアドレスはレジスタ156にシフトされる、等々である。n個のシフト・サイクルがあって、第n番目のシフト・サイクル中、レジスタ156からのデータ及びアドレス（FIFO154からのデータ及びアドレスである）はバス176にシフトされる。コントローラ152はシフト/ロード信号を駆動することによって、或は、レジスタ156-156と結合されているシフト・クロック信号（不図示）を生成することによってそれらシフト・サイクルを具現化し得る。更には、対応するFIFO154-154がコントローラ152がレジスタにロードした際に空であるので、レジスタ156-156の内の1つが特定のシフト動作中に空であれば、コントローラはその空レジスタを迂回し得て、よって、スル・データ及びマル・アドレスのバス176へのシフトを回避することによってシフト動作を短縮する。  
30

## 【0124】

図5及び図6で参照されるように、本発明の実施例に従えば、インターフェース144はインターフェース142と類似し、インターフェース132も該インターフェース13  
50

2が1つの書き込みチャネル150のみを含むことを除いてインターフェース142と類似している。

【0125】

図7は、本発明の実施例に従った図5のインターフェース140のブロック線図である。図5と連携して先に議論されたように、インターフェース140はDPSRAM100から入力データを読み取って、そのデータをハードウェアに組み込まれた74,-74に転送する。以下に議論されるように、インターフェース140の構造はデータ「ボトルネック」を低減或はなくし、パイプライン回路80(図5)がPLIICである場合、PLIICのローカル及びグローバルのルーティング・リソースを効率的に利用する。

【0126】

インターフェース140は、各ハードウェアに組み込まれたパイプライン74,-74(図5)に対して1つずつのチャネルである読み取りチャネル190,-190とコントローラ192とを含む。図示の簡略化のため、読み取りチャネル190が以下に議論されているが、他の読み取りチャネル190,-190の動作及び構造は別段の説明がない限り類似していることを理解していただきたい。

【0127】

チャネル190は FIFO194及びアドレス/識別子(ID)レジスタ196を含む。以下に議論されるように、識別子は、データを受信すべくDPSRAM100の特定箇所からそのデータを読み取る要求を為すパイプライン74,-74を識別する。

【0128】

FIFO194は2つのサブFIFO(不図示)を含み、一方がパイプライン74が入力データを読み取ることを想んでいるDPSRAM100内の箇所のアドレスを記憶し、他方がDPSRAM100から読み取られたデータを記憶する。それ故に、FIFO194は、もしパイプライン74が、コントローラ192が先行するデータの読み取りを終了するまで、チャネル190に読み取りアドレスを提供するために「待機」しなければならない場合、或は、コントローラが、コントローラが引き続くデータを読み取る前にパイプライン74が読み取りデータを検索するまで待機しなければならない場合に生じ得るボトルネックを低減或はなくする。

【0129】

FIFO194はバス198を介してパイプライン74から読み取りアドレスを受信し、そのアドレス及びIDをバス200を介してレジスタ196に提供する。IDはパイプライン74と対応して典型的には変化しないので、FIFO194はそのIDを記憶し得て、そのIDをアドレスとつなぎ合わせる。代替的には、パイプライン74はIDをバス198を介してFIFO194に提供し得る。更には、FIFO194はライン202を介してパイプライン74から準備書き込みFIFO信号を受信し、ライン204を介してCLOCK信号を受信し、(読み取りアドレスの) FIFO充満信号をライン206を介してパイプラインに提供する。加えて、FIFO194はライン208を介してコントローラ192から書き込み/読み取りFIFO信号(WRITE/READ FIFO信号)を受信し、FIFO空信号をライン210を介してコントローラに提供する。更には、FIFO194はバス212を介してコントローラ192から読み取りデータ及び対応するIDを受信し、そのデータをバス214を介してパイプライン74に提供する。パイプライン回路80(図5)がPLIICである場合、バス198, 200, 214及びライン202, 204, 206, 208, 210は好ましくはローカル・ルーティング・リソースを用いて形成され、バス212は典型的にはグローバル・ルーティング・リソースを用いて形成される。

【0130】

レジスタ196は読み取られるべき箇所のアドレスと対応するIDをバス206を介してFIFO194から受信し、そのアドレスをアドレス・バス216を介してDPSRAM100(図5)のポート108に提供して、そのIDをバス218を介してコントローラ192に提供する。更には、レジスタ196も、以下に議論されるように、アド

10

20

30

40

50

レス／IDバス220<sub>1</sub>を介してレジスタ196<sub>1</sub>～196<sub>2</sub>からアドレス及びIDを受信する。加えて、レジスタ196<sub>1</sub>はライン222を介してコントローラ192からシフト／ロード信号を受信する。パイプライン回路80（図5）がPLICである場合、バス216は典型的にはグローバル・ルーティング・リソースを用いて形成され、バス220<sub>1</sub>～220<sub>2</sub>及びライン222は好みしくはローカル・ルーティング・リソースを用いて形成される。

#### 【0131】

FIFO空信号の受信、書き込み／読み取りFIFO及びシフト／ロード信号の生成、並びに、読み取りデータ及び対応するIDの提供に加えて、コントローラ192はバス224を介してDPSRAM100（図5）のポート108から読み取られたデータを受信し、読み取りDPSRAM信号（READ DPSRAM信号）をライン226上に生成するが、そのラインがこの信号をポート108と結合する。パイプライン回路80（図5）がPLICの場合、バス224及びライン226は典型的にはグローバル・ルーティング・リソースを用いて形成される。  
10

#### 【0132】

更に図7で参照されるように、インターフェース140の動作が議論されている。

#### 【0133】

先ず、FIFO194<sub>1</sub>は、読み取りアドレスに対するFIFOの現行状態（「充満」或は「非充満」）と対応する論理レベルまでFIFO充満信号を駆動する。即ち、もしFIFO194<sub>1</sub>が読み取られるべきアドレスで充満していれば、FIFO充満の論理レベルを1つのレベルまで駆動し、もしFIFOが読み取りアドレスで充満していなければ、FIFO充満の論理レベルを別のレベルまで駆動する。  
20

#### 【0134】

次に、もしFIFO194<sub>1</sub>が読み取りアドレスで充満していなく且つパイプライン74<sub>1</sub>がより多くの入力データを処理すべく準備がされると、パイプラインは読み取られるべきデータのアドレスをバス198<sub>1</sub>まで駆動し、読み取り／書き込みFIFO信号（READ/WRITE FIFO信号）を書き込みレベルまでアサートし、よってアドレスをFIFOにロードする。図5と連携して先に議論されたように、パイプライン74<sub>1</sub>はシーケンス・マネージャ148を介して入力データ・キュー122からアドレスを得る。しかしながらFIFO194<sub>1</sub>が読み取りアドレスで充満していれば、パイプライン74<sub>1</sub>は読み取りアドレスのローディング前にFIFOが充満しなくなるまで待機する。  
30

#### 【0135】

次いで、FIFO194<sub>1</sub>はFIFO空信号を読み取りアドレスに対するFIFOの現行状態（「空」或は「非空」）と対応する論理レベルまで駆動する。即ち、もしFIFO194<sub>1</sub>が少なくとも1つの読み取りアドレスでロードされれば、FIFO空の論理レベルを1つのレベルまで駆動し、もしFIFOが読み取りアドレス無しでロードされれば、FIFO空の論理レベルを別のレベルまで駆動する。

#### 【0136】

次に、もしFIFO194<sub>1</sub>が空でなければ、コントローラ192は書き込み／読み取りFIFO信号を読み取り論理レベルまでアサートし、シフト／ロード信号をロード論理レベルまで駆動し、よってFIFOから最初にロードされたアドレスとIDをレジスタ196<sub>1</sub>にロードする。  
40

#### 【0137】

チャネル190<sub>1</sub>～190<sub>2</sub>は同様の方式で動作して、コントローラ192がFIFO194<sub>2</sub>～194<sub>3</sub>から最初にロードされたアドレスとIDをレジスタ196<sub>1</sub>～196<sub>2</sub>にそれぞれロードする。もしFIFO194<sub>2</sub>～194<sub>3</sub>の全てが空であれば、コントローラ192は前進の前にアドレスを受信すべくFIFOの内の少なくとも1つを待機する。

#### 【0138】

次いで、コントローラ192はシフト／ロード信号をシフト論理レベルまで駆動し、読み取りDPSRAM信号をアサートしてレジスタ196<sub>1</sub>～196<sub>2</sub>からアドレス及びID  
50

バス216及び218にアドレス及びIDを順次シフトして、バス224を介してDPSRAM100の対応する箇所からデータを順次読み取る。

【0139】

次に、コントローラ192は受信されたデータと対応するID（そのIDは FIFO194<sub>1</sub>～194<sub>2</sub>の各々にそれがデータの意図された受取人であるか否かを決定させる）をバス212まで駆動し、書き込み／読み取りFIFO信号を書き込みレベルまで駆動し、よってそのデータを各 FIFO194<sub>1</sub>～194<sub>2</sub>に順次書き込む。

【0140】

次いで、ハードウェアに組み込まれたバイ二ライン74<sub>1</sub>～74<sub>2</sub>はそれらの読み取り／書き込みFIFO信号を読み取りレベルまで順次アサートし、そのデータをバス214<sub>1</sub>～214<sub>2</sub>を介して順次読み取る。  
10

【0141】

更に図7で参照されるように、それらのデータ読み取り演算子のより詳細な議論が提示される。

【0142】

第1シフト・サイクル中、コントローラ192はレジスタ196<sub>1</sub>からバス216及び218までアドレス及びIDをそれぞれシフトし、読み取りDPSRAMをアサートし、そうして、バス224を介してDPSRAM100の対応する箇所からデータを読み取り、バス218からIDを読み取る。次に、コントローラ192は書き込み／読み取りFIFO信号をライン208<sub>1</sub>上で書き込みレベルまで駆動し、受信されたデータとIDをバス212に駆動する。IDが FIFO194<sub>1</sub>からのIDであるので、FIFO194<sub>1</sub>はそのIDを認識し、よって書き込み／読み取りFIFO信号の書き込みレベルに応じてバス212からデータをロードする。残りの FIFO194<sub>2</sub>～194<sub>2</sub>は、バス212上のIDがそれらのIDと対応しないので、データをロードしない。次いで、バイ二ライン74<sub>1</sub>は読み取り／書き込みFIFO信号をライン202<sub>1</sub>上で読み取りレベルまでアサートし、バス214<sub>1</sub>を介して読み取りデータを検索する。また、第1シフト・サイクル中、レジスタ196<sub>2</sub>からのアドレス及びIDはレジスタ196<sub>1</sub>にシフトされ、レジスタ196<sub>3</sub>（不図示）からのアドレス及びIDはレジスタ196<sub>2</sub>にシフトされ、等々である。代替的には、コントローラ192はIDを認識し、書き込み／読み取りFIFO信号だけをライン208<sub>1</sub>上で書き込みレベルまで駆動する。これはIDを FIFO194<sub>1</sub>～194<sub>2</sub>まで送信すべきコントローラ192に対する必要性をなくする。別の代替例において、書き込み／読み取りFIFO信号は読み取り信号のみであり、FIFO194<sub>1</sub>（他の FIFO194<sub>2</sub>～194<sub>2</sub>と共に）はバス212上のIDが FIFO194<sub>1</sub>のIDと符合する場合にバス212上にデータをロードする。これは書き込み信号を生成すべきコントローラ192の必要性をなくする。  
20

【0143】

第2シフト・サイクル中、レジスタ196<sub>1</sub>からのアドレス及びIDはバス216及び218にシフトされて、コントローラ192が FIFO194<sub>2</sub>によって特定されるDPSRAM100の箇所からデータを読み取るように為す。次に、コントローラ192は書き込み／読み取りFIFO信号を書き込みレベルまで駆動し、受信されたデータとIDをバス212に駆動する。IDが FIFO194<sub>2</sub>からのIDであるので、FIFO194<sub>2</sub>はそのIDを認識し、そうしてバス212からデータをロードする。残りの FIFO194<sub>1</sub>及び194<sub>2</sub>～194<sub>2</sub>は、バス212上のIDがそれらのIDと対応しないのでデータをロードしない。次いで、バイ二ライン74<sub>2</sub>はその読み取り／書き込みFIFO信号を読み取りレベルまでアサートし、バス214<sub>2</sub>を介して読み取りデータを検索する。また第2シフト・サイクル中、レジスタ196<sub>2</sub>からのアドレス及びIDはレジスタ196<sub>1</sub>にシフトされ、レジスタ196<sub>3</sub>（不図示）からのアドレス及びIDはレジスタ196<sub>2</sub>にシフトされ、等々である。  
40

【0144】

これはn個のシフト・サイクルの間続行、即ち、レジスタ196<sub>n</sub>（FIFO194<sub>n</sub>か  
50

らのアドレス及びID)からのアドレス及びIDがバス216及び218にそれぞれシフトされるまで続行する。コントローラ192はシフト/ロード信号を駆動することによって、或はレジスタ196<sub>1</sub>~196<sub>n</sub>と結合されたシフト・クロック信号(不図示)を生成することによってそれらシフト・サイクルを具現化し得る。更には、もし特定シフト・サイクル中にレジスタ196<sub>1</sub>~196<sub>n</sub>の内の1つが、その対応する FIFO 194<sub>1</sub>~194<sub>n</sub>が空であるため空であれば、コントローラ192は空レジスタを巡回し、よってメル・アドレスのバス216へのシフトを回避することによってシフト動作を短縮する。

## 【0145】

図5及び図6で参照されるように、本発明の実施例に従えば、インターフェース144はインターフェース140と類似し、インターフェース136も該インターフェース136が1つのみの読み取りチャネル190を含むことを除けばインターフェース140と類似し、よって1D回路を全く含まない。

10

20

## 【0146】

図8は、本発明の別の実施例に従った図4のパイプライン・ユニット230の概略プロック線図である。このパイプライン・ユニット230は、該パイプライン・ユニット230が多数のパイプライン回路80を含むことを除いて(ここでは2つのパイプライン回路80a及び80b)図4のパイプライン・ユニット78と類似している。パイプライン回路80の数を増大することは、典型的には、ハードウェアに組み込まれたパイプライン74<sub>1</sub>~74<sub>n</sub>の数nに関する増大を可能とし、よってパイプライン・ユニット78と比較してパイプライン・ユニット230の機能に関する増大を可能とする。

30

## 【0147】

図8のパイプライン・ユニット230において、サービス構成要素、即ち、通信インターフェース82、コントローラ86、例外マネージャ88、コンフィギュレーション・マネージャ90、並びに、任意選択的な工業規格バス・インターフェース91はパイプライン回路80a上に配置され、パイプライン74<sub>1</sub>~74<sub>n</sub>及び通信シェル84はパイプライン回路80b上に配置される。サービス構成要素及びパイプライン74<sub>1</sub>~74<sub>n</sub>を別々のパイプライン回路上に位置決めすることによって、サービス構成要素及びパイプラインが同一のパイプライン回路上に位置決めされている場合に可能であるものよりも、より多くの数nのパイプライン及び/或はより複雑なパイプラインを含み得る。代替的には、パイプライン74<sub>1</sub>~74<sub>n</sub>をインターフェース82にインターフェースする通信シェル84の部分とコントローラ86はパイプライン回路80a上に配置される。

## 【0148】

図9は、本発明の実施例に従った図8のパイプライン・ユニット230のパイプライン回路80a、80b及びデータ・メモリ92の概略プロック線図である。図9のパイプライン構成要素が2つのパイプライン回路上に配置されていること以外、パイプライン回路80a及び80bとメモリ92の構造及び動作は、図5のパイプライン回路80及びメモリ92のものと同一である。

40

## 【0149】

先行する議論は当業者が本発明を作製し使用することを可能とすべく提示されている。種々実施例への様々な変更は当業者には容易に明かであろうし、ここでの包括的な原則は本発明の精神及び範囲から逸脱することなしに他の実施例及び適用例に適用され得る。よって、本発明は図示された実施例に限定されることが意図されておらず、ここに開示された原理及び特徴と一貫した最も広い範囲と一致されるべきものである。

## 【前面の簡単な説明】

## 【0150】

【図1】図1は、従来の多数プロセッサ・アーキテクチャを有する計算マシンのプロック線図である。

【図2】図2は、従来のハードウェアに組み込まれたパイプラインのプロック線図である。

【図3】図3は、本発明の実施例に従ったピア-ベクトル・アーキテクチャを有する計算

50

マシンのブロック線図である。

【図4】図4は、本発明の実施例に従った図3のパイプライン加速器のブロック線図である。

【図5】図5は、本発明の実施例に従った図4のハードウェアに組み込まれたパイプライン回路とデータ・メモリとのブロック線図である。

【図6】図6は、本発明の実施例に従った図5の通信シェルのメモリ書き込みインターフェースのブロック線図である。

【図7】図7は、本発明の実施例に従った図5の通信シェルのメモリ読み取りインターフェースのブロック線図である。

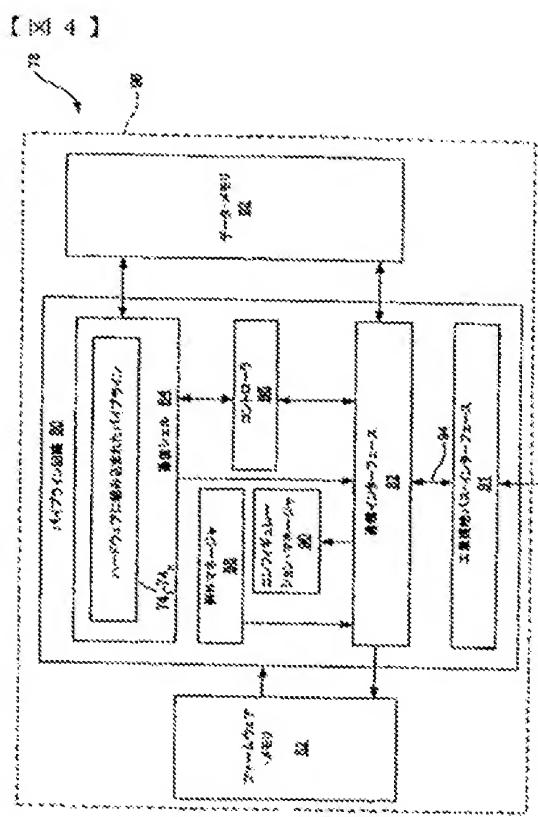
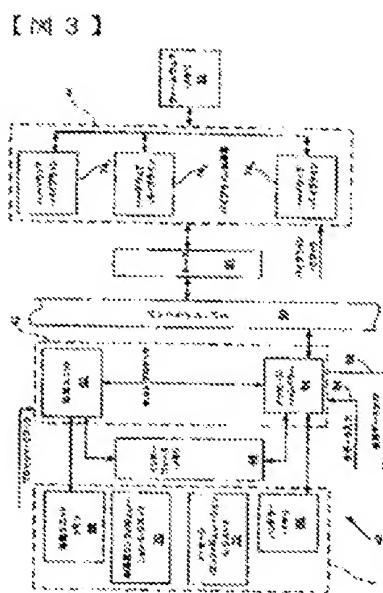
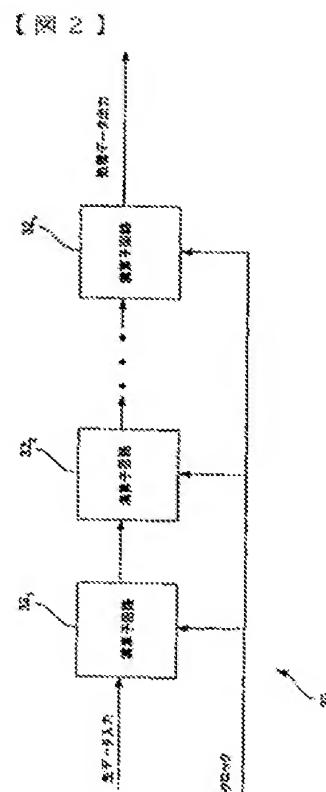
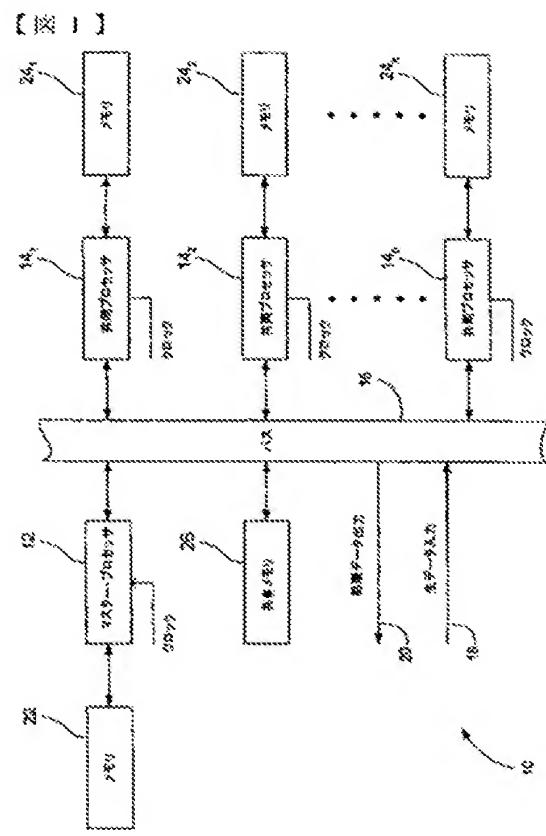
【図8】図8は、本発明の別の実施例に従った図3のパイプライン加速器のブロック線図である。

【図9】図9は、本発明の実施例に従った図8のハードウェアに組み込まれたパイプライン回路とデータ・メモリとのブロック線図である。

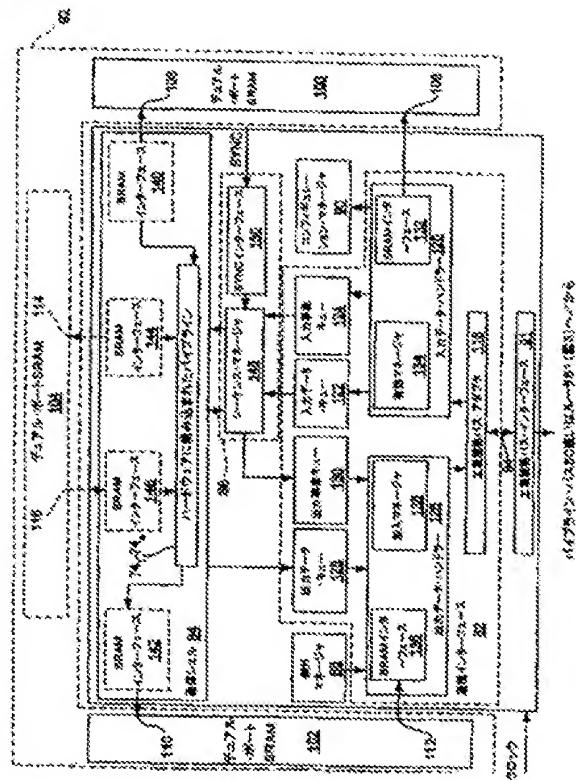
【符号の説明】

【0151】

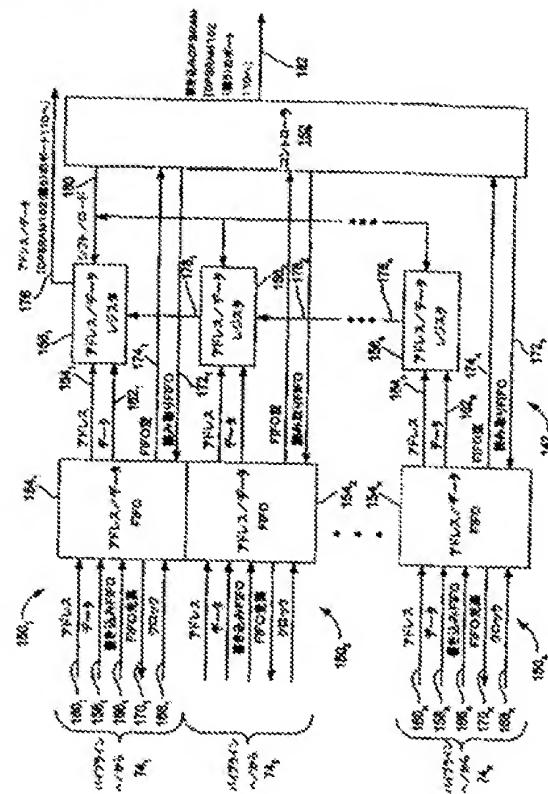
1 0	計算マシン	20
1 4	共同プロセッサ	
4 0	ピア-ペクトル・マシン	
4 2	ホストプロセッサ	
4 4	パイプライン加速器	
4 6	プロセッサ・メモリ	
4 8	インターフェース・メモリ	
5 0	パイプライン・バス	
5 2	ファームウェア・メモリ	
5 4	生データ入力ポート	
5 8	処理データ出力ポート	
6 1	ルータ	
6 2	処理ユニット	
6 4	メッセージ・ハンドラー	
6 6	処理ユニット・メモリ	30
6 8	ハンドラー・メモリ	
7 0	加速器コンフィギュレーション・レジストリ	
7 2	メッセージ・コンフィギュレーション・レジストリ	
7 4	ハードウェアに組み込まれたパイプライン	
7 8	パイプライン・ユニット	
8 0	パイプライン回路	
8 6	パイプライン・コントローラ	
8 8	例外マネージャ	
9 0	コンフィギュレーション・マネージャ	
9 1	工業規格バス・インターフェース	40



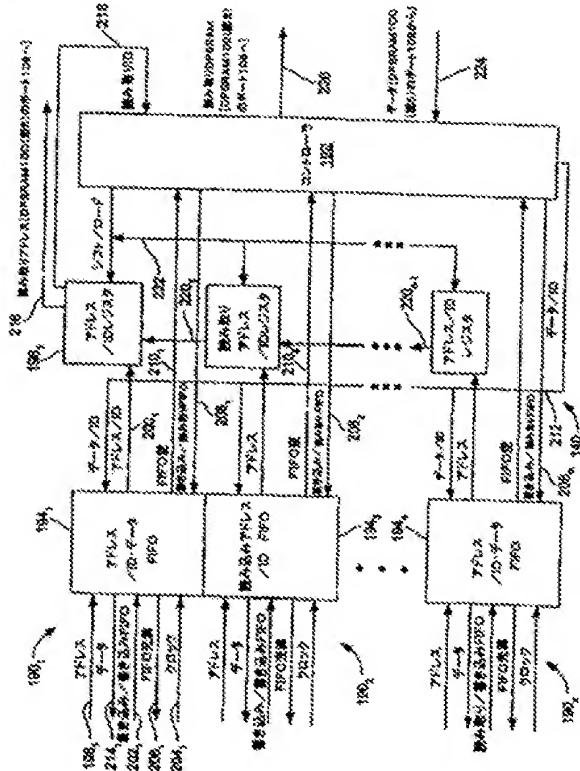
13351



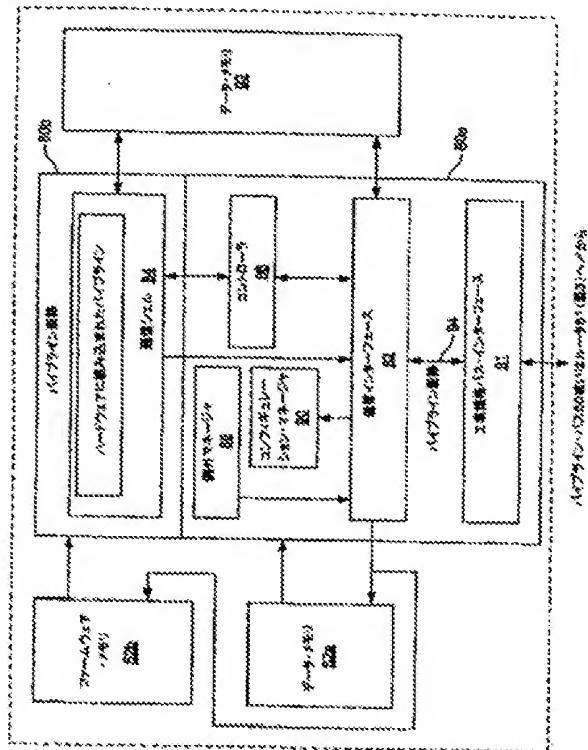
[ 6 ]



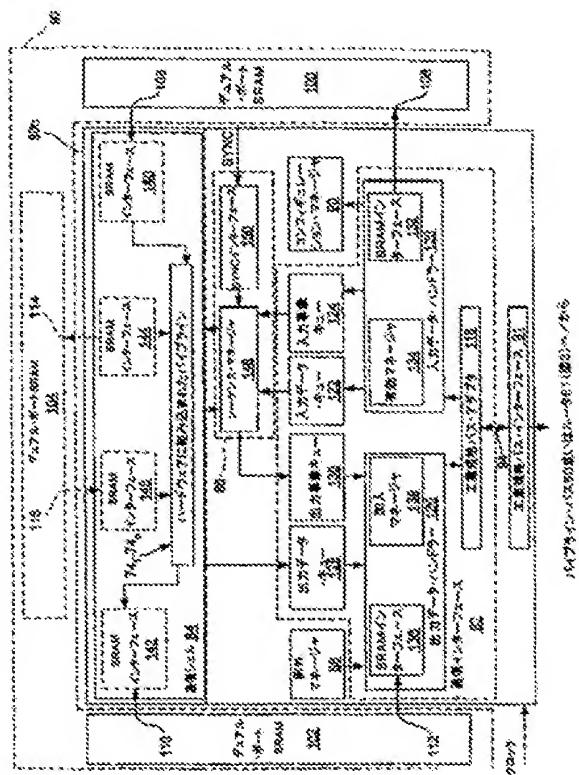
〔四七〕



[ FIG. 8 ]



〔四〇〕



## 【国際調査報告】

## INTERNATIONAL SEARCH REPORT

PCT/US 03/34558

A. CLASSIFICATION OF SUBJECT MATTER		
IPC 7    G06F9/46    G06F9/38    G06F15/78		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols)		
IPC 7    G06F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronical data base consulted during the international search (name of data base and, where practical, search terms used)		
EPO-Internal		
C. DOCUMENTS CONSIDERED TO BE IRRELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Applicant's ref. No.
X	<p>EP 1 661 439 A (HEWLETT PACKARD CO) 26 December 2000 (2000-12-28) page 4 paragraph [0018] - paragraph [0021] paragraph [0028] - paragraph [0031] paragraph [0054] - paragraph [0058]</p> <p>-----</p> <p>US 5 892 962 A (CLOUTIER ET AL) 6 April 1999 (1999-04-06)</p> <p>claim 7 figures 1,2 column 2, line 53 - line 61 column 3, line 34 - column 4, line 6 column 5, line 24 - line 29</p> <p>-----</p> <p>-----</p>	1-14, 41-50
X		1-8, 41-43, 49,50
<input checked="" type="checkbox"/> Further documents are listed in the continuation of box C. <input type="checkbox"/> Patent family members are listed in annex.		
* Special categories of cited documents:		
<p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"B" earlier document but published on or after the International filing date</p> <p>"C" document which may throw doubt on novelty, soundness or which is cited to establish the publication date of another document or other special reason (see "Explanation")</p> <p>"D" document relating to an oral disclosure, use, exhibition or other means</p> <p>"P" document published later than the International filing date but later than the priority date claimed</p>		
Date of the actual completion of the International search		Date of mailing of the International search report
18 February 2005		16 06 2005
Name and mailing address of the ISA European Patent Office, P.O. 5815 Potsdamer 2 80 2290 Munich 80 Tel. (+49-89) 269-0045, Te. 31 851 800 01 Fax. (+49-89) 269-3616		Authorized officer Bosch Vivancos, P

Form PCT/ISA/2000 (second sheet last page)

page 1 of 2

## INTERNATIONAL SEARCH REPORT

PCT/US 03/34558

## C. (CONTINUATION) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Reference to claim No.
A	SALCIC 2 ET AL: "FLIX environment for generation of custom-configurable machines in FPLDs for embedded applications" MICROPROCESSORS AND MICROSYSTEMS, IPC BUSINESS PRESS LTD, LONDON, GB, vol. 23, no. 8-9, 15 December 1999 (1999-12-15), pages 513-526, XP006254077 ISSN: 0141-9331 page 520, right-hand column, line 1 - page 521, left-hand column, line 10 .....	1-14, 41-50

Form PCT/US/AS/10 (continuation of second sheet) (January 2004)

page 2 of 2

## INTERNATIONAL SEARCH REPORT

PCT/US 03/34558

## Box I Observations where certain claims were found unsearchable (Continuation of Item 1 of first sheet)

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1.  **Claims Nos.:** because they relate to subject matter not required to be searched by the Authority, namely:
  
  
  
2.  **Claims Nos.:** because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:
  
  
  
3.  **Claims Nos.:** because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 9.4(a).

## Box II Observations where unity of invention is lacking (Continuation of Item 2 of first sheet)

This International Searching Authority found multiple inventions in this International application, as follows:

see additional sheet

1.  As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.
  
2.  As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.
  
3.  As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:
  
  
  
4.  No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

## Remarks on Protest

The additional search fees were accompanied by the applicant's protest.  
 No protest accompanied the payment of additional search fees.

International Application No. PCT/ US 03/34558

## FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1,2

A two integrated-circuit pipeline accelerator.

1.1. claim: 3

A reconfigurable pipeline accelerator

1.2. claims: 4-14,41-50

Method and apparatus for pipelining data communications and processing in a processor/coprocessor system.

\*\*\*

2. claims: 15,31-34,59

Method and apparatus for configuring a hardwired pipeline.

\*\*\*

3. claims: 16,35-40,60

Exception handling in a processor/co-processor system

\*\*\*

4. claims: 17-23,51-54

Method and apparatus for routing data in a processor/co-processor system.

\*\*\*

5. claim: 24

A multithreaded coprocessor.

\*\*\*

6. claims: 26-30,55-58

Control of the request handling order in a co-processor.

\*\*\*

7. claims: 61-65

A method for designing hardwired pipelines.

\*\*\*

## INTERNATIONAL SEARCH REPORT

on patent family members

PCT/US 03/34558

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
EP 1061439	A	20-12-2000	EP 1061439 A1 EP 1184562 A1 WO 0377627 A1 JP 2003502728 T US 6782445 B1	20-12-2000 06-06-2001 21-12-2000 21-01-2003 24-08-2004
US 5892962	A	06-04-1999	CA 2215598 A1	12-05-1998
-----				

Form PCT/2004/01 for patent family search (January 2005)

## フロントページの続き

(31)優先権主張番号 10/684,053

(32)優先日 平成15年10月9日(2003.10.9)

(33)優先権主張国 米国(US)

(31)優先権主張番号 10/684,057

(32)優先日 平成15年10月9日(2003.10.9)

(33)優先権主張国 米国(US)

(31)優先権主張番号 10/684,102

(32)優先日 平成15年10月9日(2003.10.9)

(33)優先権主張国 米国(US)

(81)指定国 AP(88, GH, GR, KE, LS, MW, MZ, SD, SI, SZ, TZ, BG, ZR, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), EP(AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OA(BF, BJ, CF, CG, CI, CR, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), AE, AG, AI, AR, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GR, HR, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LR, LS, LT, LU, LV, MA, MD, MG, MR, MN, MW, MX, MZ, NI, NO, NZ, OR, PG, PL, PT, RD, RU, SC, SD, SE, SG, SK, SI, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW

(74)代理人 100135585

弁理士 西尾 務

(72)発明者 ラーブ, ジョン, ダブリュ,

アメリカ合衆国 バージニア州 20110 マナサス, リバー クレスト ロード 9390

(72)発明者 ジャクソン, ラリー

アメリカ合衆国 バージニア州 20112 マナサス, クレストロック ドライブ 130  
93

(72)発明者 ジョウンズ, マーク

アメリカ合衆国 バージニア州 20120 セントレビル, オークマー ブレイス 15342

(72)発明者 カーサロ, トロイ

アメリカ合衆国 バージニア州 22701 カルペニー, ケストラル モート 1524